DISPOZITIV PENTRU HRĂNIREA PUILOR DE GĂINĂ

Student DANIEL BĂDESCU,
Craiova

Tinerilor de la sate le propun construcția unui dispozitiv care le poate fi de un real folos: el asigură distribuirea hranei (sub formă de boabe sau uruială) într-o cantitate bine determinată, dinainte prestabilită.

Ideea care stă la baza construcției, simplitatea ei o fac aptă de a fi aplicată și în alte sectoare de activitate prin reproiectarea și redimensionarea ei. Construit la scară mai mică, dispozitivul poate asigura hrănirea peștișorilor dintr-un acvariu, iar la scară mai mare poate asigura cu succes dozarea unei componente a amestecului de furaje pentru animale.

Dimensiunile date în schițe corespund dispozitivului destinat hrănirii puilor de găină, care folosesc hrana sub formă de uruială.

A. REALIZARE ȘI MOD DE FUNCȚIONARE

Cantitatea de uruială necesară puilor este depozitată în coșul (5). Accesul puilor la uruială în restul timpului este împiedicat de capacul (6). Uruiala este eliberată prin mecanismul de obturare dispus la partea de jos a coșului, alcătuit din reperele (7), (8), (9) (fig. 1).

Acționarea mecanismului de obturare se face electromagnetic, cu ajutorul electromagnetului (3). Comanda electromagnetului este asigurată de un circuit electronic de temporizare, care asigură timpul de deschidere a obturatorului. Cantitatea de hra-

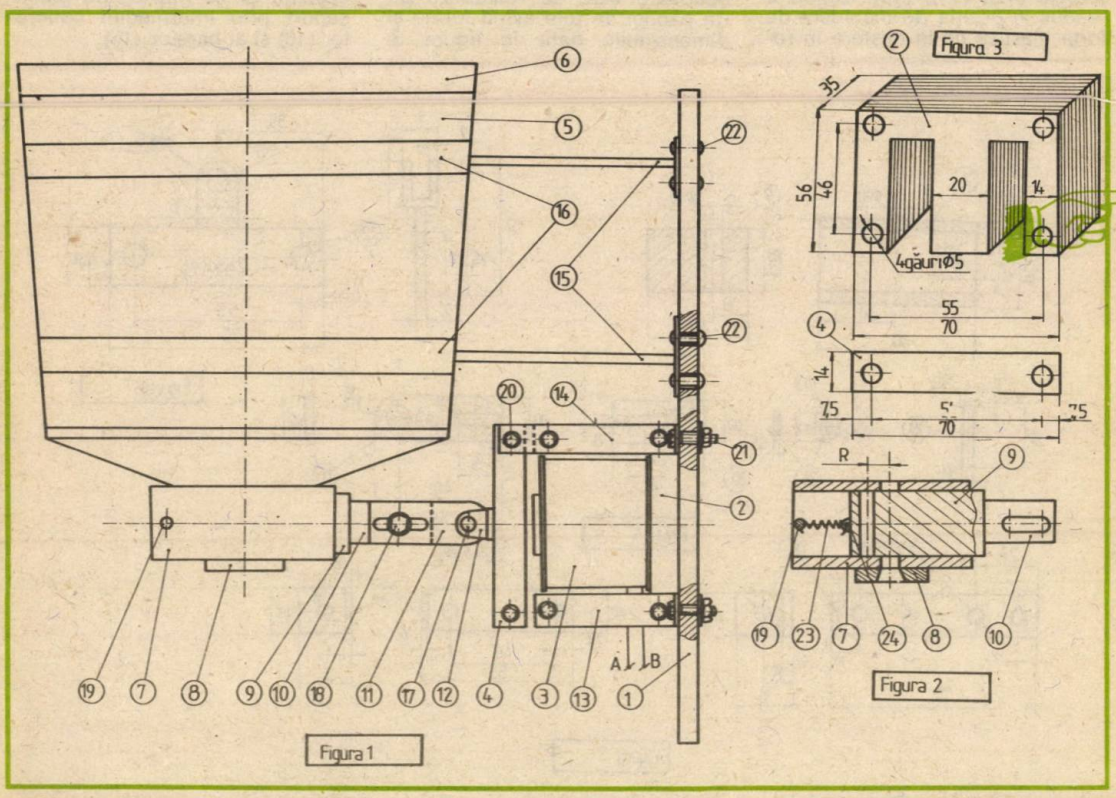


Figura 1

Figura 2

Figura 3

nă este direct proporţională cu timpul cît obturatorul este deschis, acest timp putînd fi reglat din circuitul de temporizare.

Mecanismul de obturare (fig. 2) funcţionează pe baza coincidenţei sau necoincidenţei axelor geometrice ale alezajelor pieselor (7) şi (9). În repaos, obturatorul este închis. În momentul începerii distribuirii hranei, circuitul de temporizare comandă închiderea electromagnetului prin intermediul releului REL.

Armătura (4) va fi atrasă, prin intermediul pieselor (12), (11), (10), asigurînd deplasarea cu distanţa R a piesei (9) în interiorul piesei (7). În acest fel se asigură coincidenţa axelor celor două alezaje, uruiala putînd să se scurgă din coş în tava montată la baza dispozitivului. După expirarea timpului de temporizare, electromagnetul eliberează armătura (4). Ansamblul, format de armătură împreună cu piesele (12), (11), (10), (9), va fi readus în poziţia de închidere de forţa elastică ce ia naştere în resortul (23). Acesta acţionează între ştiftul (19) (montat transversal în piesa 7) şi piesa (9), de care este fixat prin intermediul urechii de prindere (24). În acest moment, curgerea hranei este oprită.

Releul de timp (fig. 7) asigură temporizări pînă la maximum 4 minute. Este o schemă simplă care foloseşte ca elemente active tranzistoarele AC180. La apăsarea pe butonul B se încarcă condensatorul de 1 000 μF, în timp ce se negativează baza tranzistorului T1, aducîndu-l în regim de conducţie, ceea ce face ca să pozitiveze baza celui de-al doilea tranzistor pnp, blocîndu-l.

În colectorul tranzistorului T2 se obţine o tensiune negativă, care permite celui de-al treilea tranzistor să se deschidă, acţionînd releul REL. Acesta la rîndul lui acţionează electromagnetul. Timpul de anclanşare se reglează din cele două potenţiometre de 500 kΩ şi 100 kΩ. Se poate folosi releul de la magnetofonul TESLA B4, consumul releului fiind de cca 8 mA.

B. CONSTRUCŢIE

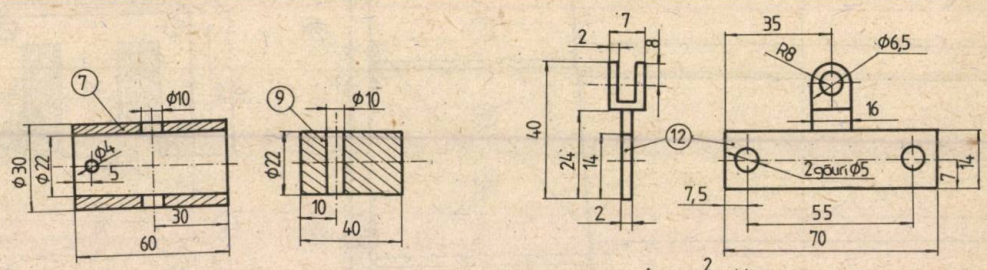
Construcţia începe prin confecţionarea electromagnetului de acţionare care se realizează pe un pachet de tole avînd forma şi dimensiunile date în figura 3. Personal am folosit un pachet de tole recuperat de la transformatorul de reţea al receptorului radio MAESTRO.

Bobina electromagnetului are 5 000 de spire CuEm ∅ 0,2 mm. Tolele E se vor asambla conform figurii 1. Piesele (13) şi (14) se confecţionează, conform figurii 5, dintr-o platbandă de oţel cu grosimea de 2 mm. Ele asigură fixarea electromagnetului pe placa-suport (1) prin intermediul celor 4 şuruburi hexagonale M8 notate (21) în figura 1.

Piesa (14) asigură şi articularea armăturii 4. Armătura (4) este realizată din pachetul de tole I, în mijlocul cărora s-a introdus piesa (12), realizată conform figurii 6. Pentru asamblarea tolelor E şi I se folosesc şuruburi M5.

Dintr-un material corespunzător se confecţionează celelalte repere, conform schiţelor respective.

Coşul (5) poate fi confecţionat dintr-o folie de tablă neagră, sau dintr-o găleată galvanizată din cele care se găsesc la magazinele cu articole de menaj. El va avea la partea inferioară o gaură cu ∅ 10 mm, a cărei axă la montare trebuie să coincidă cu axa găurilor piesei (7). Piesa (7) este sudată la partea inferioară a coşului. El se fixează pe placa-suport prin intermediul colierelor (16) şi al barelor (15).



După ce toate elementele au fost asamblate conform desenului de ansamblu, părțile metalice se protejează cu o vopsea pe bază de ulei.

Elementele aflate în mișcare se ung cu puțin ulei mineral pentru o funcționare lină și fără șocuri.

C. REGLAJ

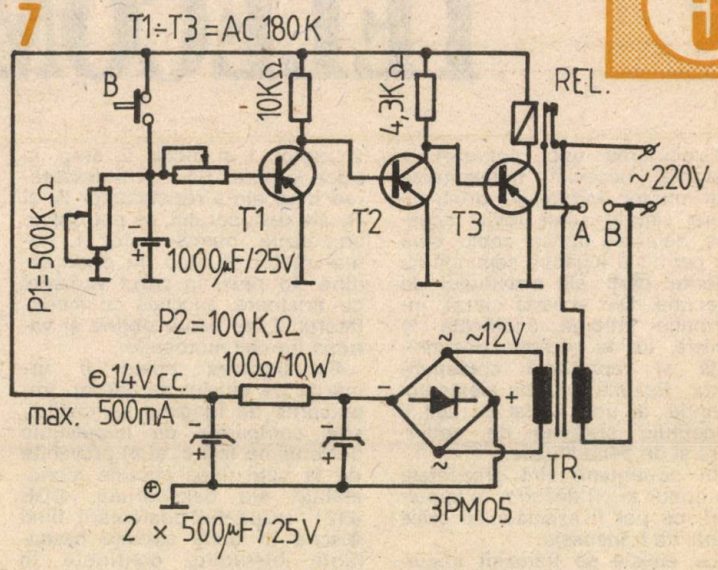
Se comandă închiderea releului, care închide electromagnetul. Se slăbește șurubul (18), acționîndu-se manual asupra piesei (9), pînă cînd axa ei coincide cu axa piesei (7) și uruiala curge din coș. Se strînge bine șurubul (18), dispozitivul fiind gata pentru utilizare.

BIBLIOGRAFIE:

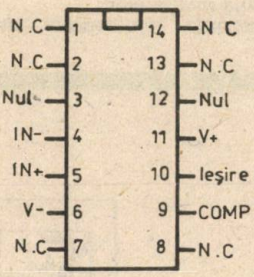
C. BISTRICEANU — Creșterea animalelor și păsărilor

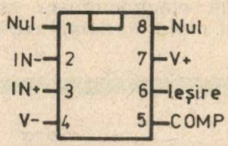
ILIE MIHĂESCU, SERGIU FLORICĂ — 101... montaje electronice.

7



memorator

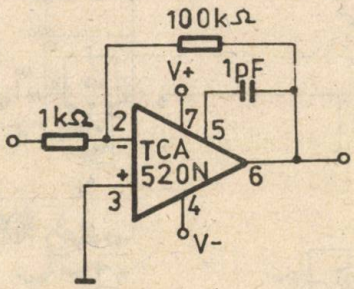




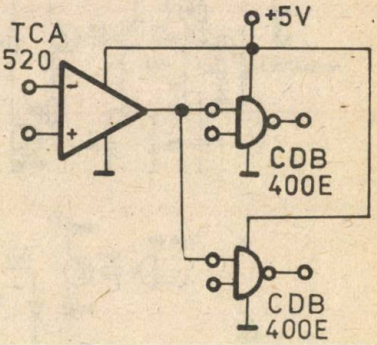
— **Comparator cu ieșire TTL.**

TCA 520 520 N

Circuitele integrate TCA520, TCA520N sînt amplificatoare operaționale destinate aplicațiilor de puteri și tensiuni reduse, precum și funcției de comparator în sisteme digitale. Aceste circuite pot fi compensate în frecvență cu un singur condensator. TCA520 și TCA520N au caracteristicile electrice identice, diferențierea fiind dată de tipul capsulei utilizate.



— **Amplificator rapid $G_v = -100$; S = 25 V/ns**



5

TELECOMANDĂ

Prof. M. TODICĂ

Executarea unor comenzi la distanță necesită transmiterea mai multor informații, printr-un canal simplu: unde radio, acustice, semnale optice, cablu, care nu permit o legătură separată cu diferite părți ale sistemului de execuție. Din această cauză, informația trebuie codificată la emisie, iar la recepție decodificată și repartizată corespunzător. Rezultă deci că elementul esențial al unui astfel de lanț îl constituie sistemul de codificare și de decodificare.

În continuare sînt prezentate un codor și un decodor în impulsuri, ce pot fi asociate cu orice canal de transmisie.

La emisie se transmit impulsuri de lungime variabilă t_1, cu pauze de lungime t_2, de frecvență foarte joasă, iar la recepție se folosește un filtru digital. Fiecare impuls reinițializează comanda astfel încît orice informație falsă este imediat corectată. Pierderea legăturii cu impulsurile de comandă aduce tot sistemul în stare de „stop".

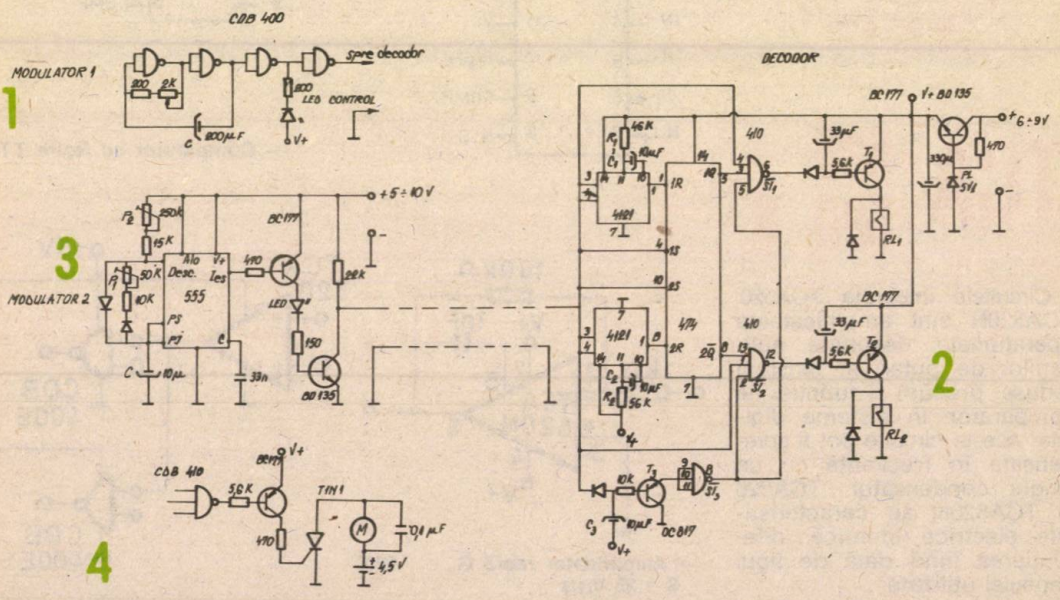
Cu acest sistem pot fi comandate două electromotoare astfel: numai motor 1, numai motor 2, motor 1 și motor 2, stop, în orice ordine. Sau, prin inversarea între ele a rezistențelor R_1 și R_2 ale decodorului, se pot obține comenzile: numai motor 1, numai motor 2, stop, în orice ordine. În plus, în cazul variantei cu tiristoare, asociată cu modulatorul 2, se poate obține și variația turației motoarelor.

Funcționarea poate fi urmărită pe tabelul de adevăr. Impulsurile de lungime variabilă t_1 sînt comparate cu impulsurile de lungime fixă τ_1 și τ_2 provenite de la cele două circuite monostabile ale decodorului, CDB 4121, rezultatul comparării fiind înscris în două circuite basculante bistabile, conținute în capsula CDB474.

Transferul informației spre ieșire are loc numai în intervalul t_2 dintre impulsuri, deoarece în intervalul t_1 porțile $\overline{SI_1}$ și $\overline{SI_2}$ sînt blocate. În felul acesta se evită transmiterea stărilor de nedeterminare ale bistabilelor din intervalul t_1. În plus, acționînd asupra lui t_2, se poate modifica timpul de conducție a tiristoarelor, deci turația motoarelor.

În prezența impulsurilor de comandă, condensatorul C_3 se încarcă, menținînd deschis cu o anumită constantă de timp tranzistorul T_3, deci nivel logic zero la intrările porții $\overline{SI_3}$ și nivel logic 1 la cîte una din intrările porților $\overline{SI_1}$ și $\overline{SI_2}$. În absența acestora, $\overline{SI_3}$ trece în starea zero, blocînd porțile $\overline{SI_1}$ și $\overline{SI_2}$. Tot sistemul trece în starea „stop".

Sînt prezentate două variante de modulatoare. Modulatorul 1, realizat cu circuitul CDB400, nu permite reglajul lui t_2, deci al turației motoarelor, și va fi asociat cu varianta cu relee a decodorului. Modulatorul 2, realizat cu circuitul βE555, permite reglajul independent al lui t_1 și t_2 și poate fi asociat și cu decodorul cu tiristoare. Legătura se realizează printr-un cablu bifilar sau optic, dacă se folosește adaptorul din figura 5. La comanda cablu, se pot folosi și impulsuri mai scurte. Pentru aceasta se vor schimba simultan, și cu valori identice, condensatoarele C_1 și C_2 ale decodorului și condensatorul C al modulatorului. La comanda optică sau radio, pentru obținerea unor impulsuri ferme,



se vor folosi relee „Reed". Aceste relee prezintă o anumită inerție mecanică, ceea ce face obligatorie folosirea impulsurilor de comandă mai lungi.

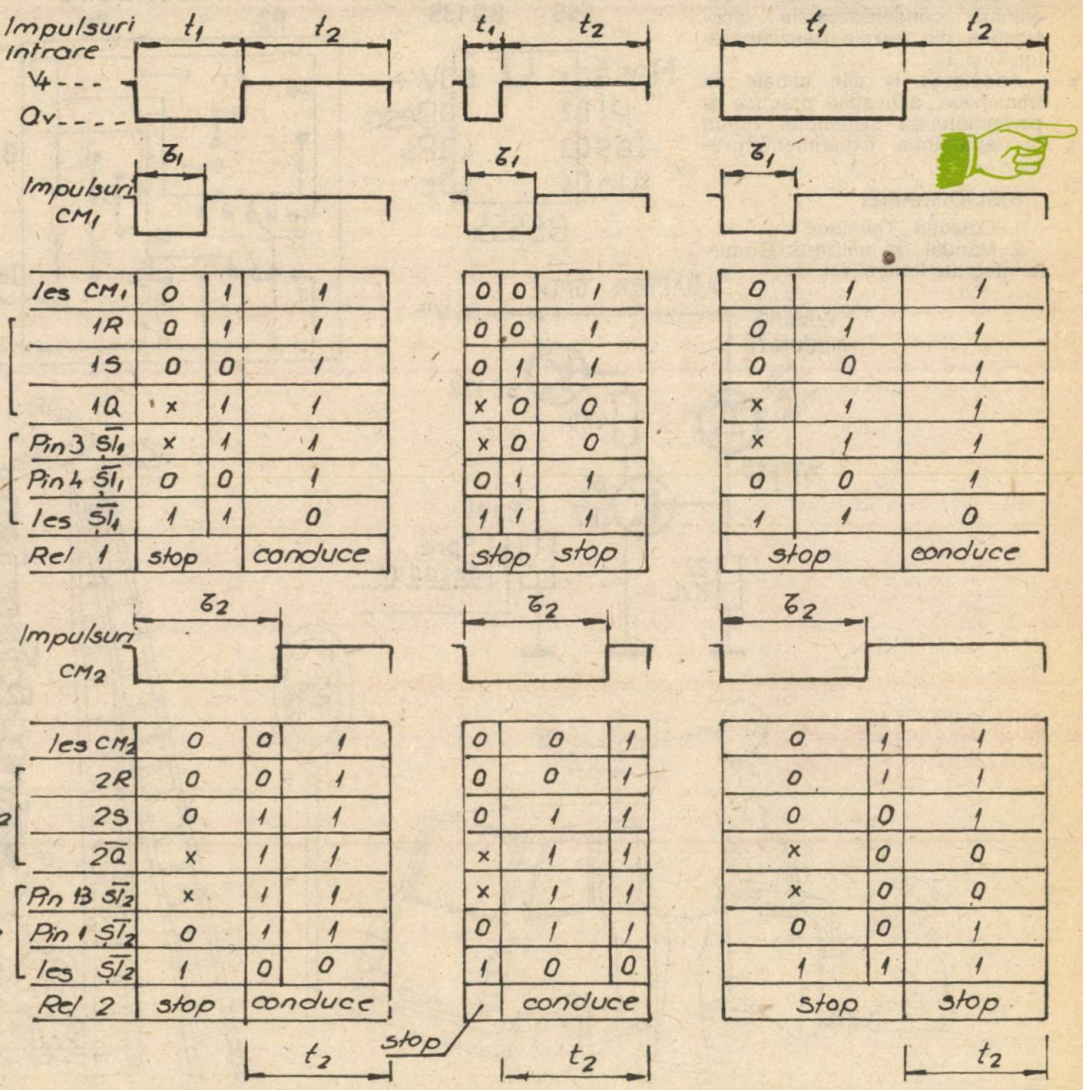
Realizare practică și reglaje. În figurile 6 și 7 sînt prezentate cablajele pentru decodor și modulatorul 2. Dacă sînt corect executate, montajele funcționează de la prima încercare. Funcționarea modulatoarelor este atestată de funcționarea LED-urilor de control. Lungimea impulsurilor t_1 și t_2 se reglează cu potențiometrele P_1 și P_2 la modulatorul 2, iar la modulatorul 1 cu ajutorul potențiometrului P. Dacă impulsurile t_1 nu sînt suficient de scurte, pentru a asigura comanda unilaterală a releului 2, se va micșora condensatorul C al modulatorului.

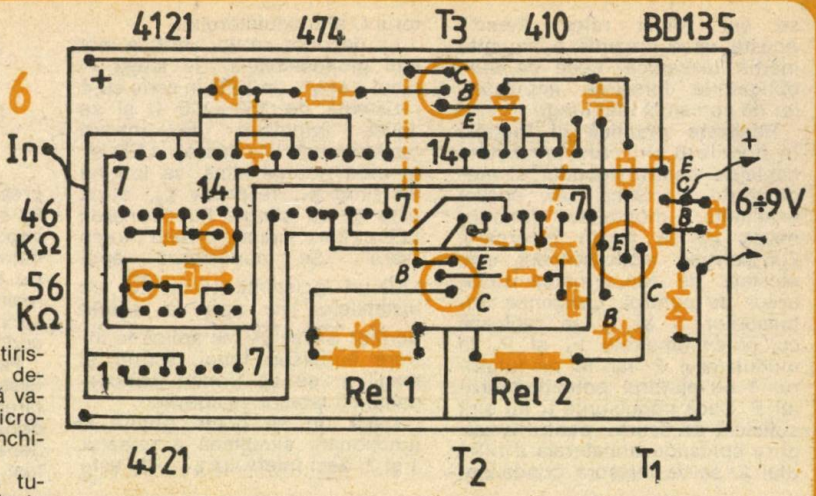
La decodor se vor verifica mai întîi monostabilele. Se leagă la pinul 1 și V+ un LED în serie cu o rezistență de 150—200 Ω și se aplică impulsuri de intrare (scurte puneri la masă). LED-ul, în mod normal stins, va lumina un timp τ_1, respectiv τ_2, după care se va stinge. Funcționarea LED-urilor indică nivelul logic „zero". Se conectează apoi LED-uri la ieșirile 1Q și $\overline{2Q}$ ale bistabilelor, iar apoi la ieșirile porților $\overline{SI}_1$ și $\overline{SI}_2$, se aplică la intrare impulsuri lungi, scurte și medii și se va urmări concordanța cu tabelul de adevăr.

Dacă nu se poate obține o funcționare simultană a releelor 1 și 2, deci intervalul $\tau_1 - \tau_2$ este prea mic, se va mări rezistența R_2 sau se va micșora R_1. Foarte important este ca impulsurile de intrare să fie dreptunghiulare. De aceea este recomandat să se realizeze mai întîi modulatorul, apoi să se treacă la testarea decodorului.

Pentru evitarea comenzilor false sînt recomandate alimentarea, cu surse separate, a decodorului și motoarelor electrice, deparazitarea acestora și, eventual, ecranarea montajului elec-

TABEL DE ADEVĂR





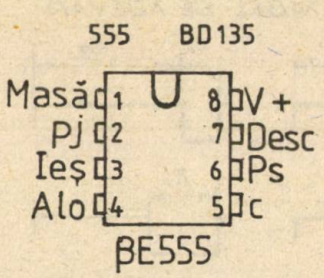
tronic. În cazul variantei cu tiristoare, condensatoarele de deparazitare nu trebuie să aibă valori prea mari, de ordinul microfarazilor, pentru a permite închiderea tiristoarelor.

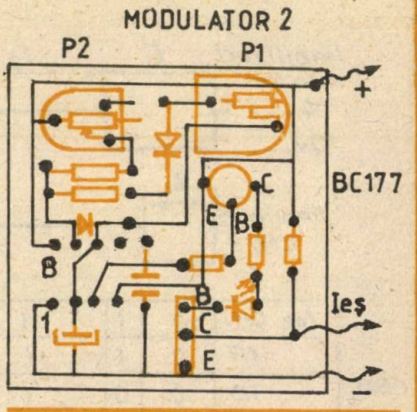
Se poate obține variația turației motoarelor și în varianta cu relee, dar, în acest caz, se vor elimina condensatoarele electrolitice din bazele tranzistoarelor T_1 și T_2.

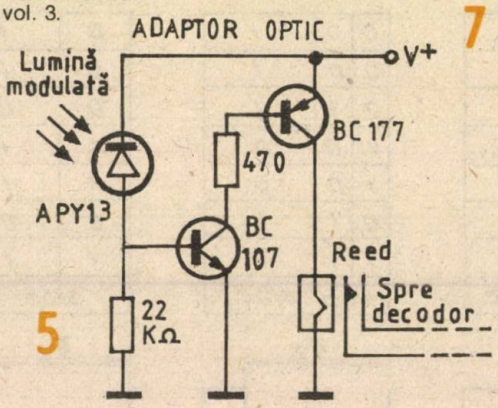
Adaptarea la alte canale de transmisie, aplicațiile practice și perfecționarea sistemului rămîn la latitudinea experimentatorului.

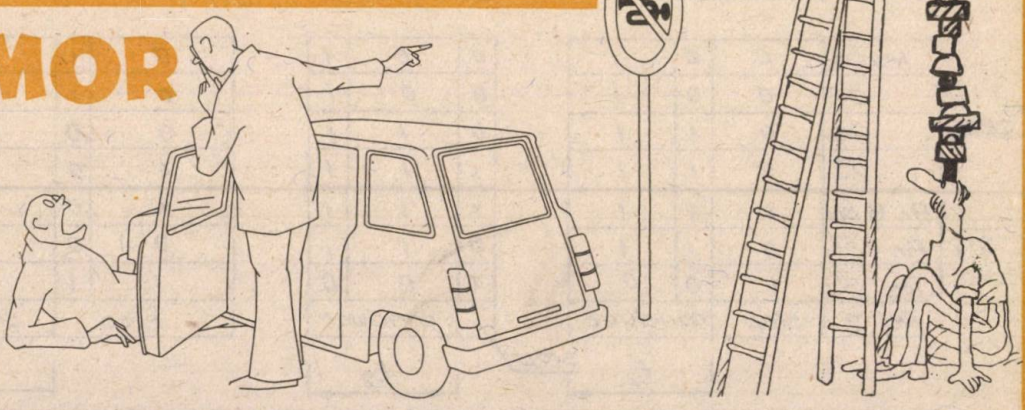
BIBLIOGRAFIE:

1. Colecția „Tehnium"
2. Manual de utilizare. Circuite integrate liniare. vol. 3.









COMANDĂ ASPECTOMAT

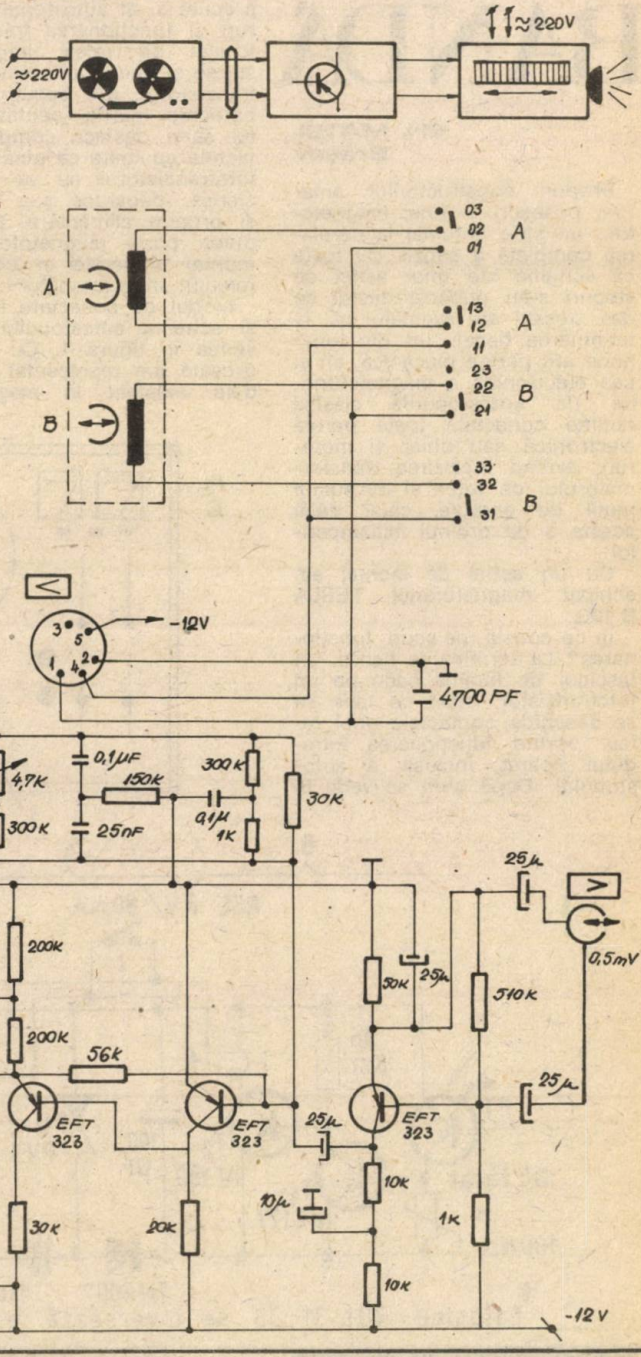
5

E. STRĂINU

Diaproiectoarele tip „Aspectomat" au posibilitatea schimbării succesive a cadrelor cu ajutorul unui cablu de telecomandă. S-a constatat însă că unii profesori întîmpină greutăți la manevrarea acestor aparate, atenția lor fiind distribuită, pe timpul expunerii, între noțiunile ce trebuie transmise și momentele în care trebuie să manevreze aparatura. Pentru eliminarea acestui neajuns propun un dispozitiv cu ajutorul căruia aspectomatul este comandat automat cu ajutorul unor semnale imprimate pe bandă de magnetofon. Lecția, expunerea sau noțiunile ce necesită exemplificări cu diapozitive sînt imprimate normal, pe două piste ale magnetofonului. Pe celelalte două piste se imprimă semnale foarte scurte, care, la momentele oportune, prin intermediul dispozitivului amintit, comandă schimbarea imaginilor.

Dispozitivul este format dintr-un etaj preamplificator cu corecție, montat pe trei tranzistoare EFT323, și un etaj final, realizat cu un tranzistor EFT323 (sau EFT170, AC180), care are ca sarcină un releu. Semnalele culese de pe bandă sînt amplificate (ele avînd o amplitudine de 0,5 mV) și la apariția lor releul cuplează pe un timp foarte scurt (0,5—1 s). Contactele releului închid circuitul motorului de deplasare a cadrelor din aspectomat. În continuare ciclul se repetă. Valorile tuturor pieselor sînt date în schemă.

În decursul utilizării sale s-a constatat eficiența ridicată a acestui dispozitiv, care duce la ușurarea expunerii lecțiilor, conferințelor etc.



5

STOP-BANDĂ

GH. MATEI,
Brașov

Propun constructorilor amatori, posesori ai unui magnetofon, un stop automat la derularea completă a benzii. Cu toate că scheme ale unor astfel de stopuri s-au publicat destul de des, acesta are avantajul că, la terminarea benzii, ies din funcțiune atît partea mecanică, cît și cea electronică a magnetofonului (la autostopurile clasice rămîne conectată toată partea electronică sau chiar și motorul), evitînd încălzirea transformatorului de rețea și consumul inutil de energie, chiar dacă acesta e de ordinul miliamperilor.

Cu un astfel de montaj am echipat magnetofonul TESLA B 100.

În ce constă, pe scurt, funcționarea? La terminarea benzii, un fascicul de lumină cade pe un fototranzistor, ceea ce face să se deschidă contactele unui releu, oprind funcționarea întregului aparat, inclusiv a autostopului. După cum se vede în schemă, atît timp cît banda oprește fasciculul de lumină să cadă pe fototranzistor, tranzistorul T1 rămîne blocat, iar T2 rămîne deschis deoarece are baza negativată și releul rămîne anclanșat. La apariția fasciculului luminos, T1 se deschide și blochează pe T2. În acest moment, contactele releului se desfac. Aceste contacte, care au închis circuitul magnetofonului, precum și al autostopului, întrerup și funcționarea transformatorului de rețea; implicit se stinge și becul care lumina fototranzistorul, dar becul, avînd o oarecare inerție, permite releului să-și desfacă complet contactele cu toate că pînă la urmă fototranzistorul nu va mai anclanșa, deoarece și-a întrerupt și propria alimentare. Pentru a putea porni magnetofonul, am montat în paralel cu contactele releului un întrerupător I.

Modul de conectare în circuit și schema autostopului se pot vedea în figura 1. Cu linie îngroșată am reprezentat circuitul deja existent în magnetofon. Partea A reprezintă circuitul de alimentare a autostopului și partea B circuitul de acționare a releului.

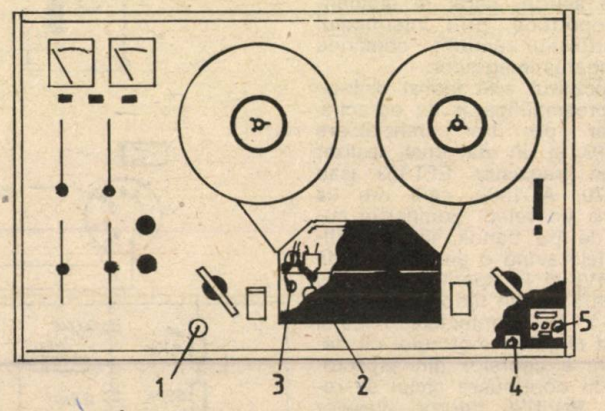
Semireglabilul P se reglează pentru o funcționare optimă a montajului.

INDICAȚII DE MONTAJ

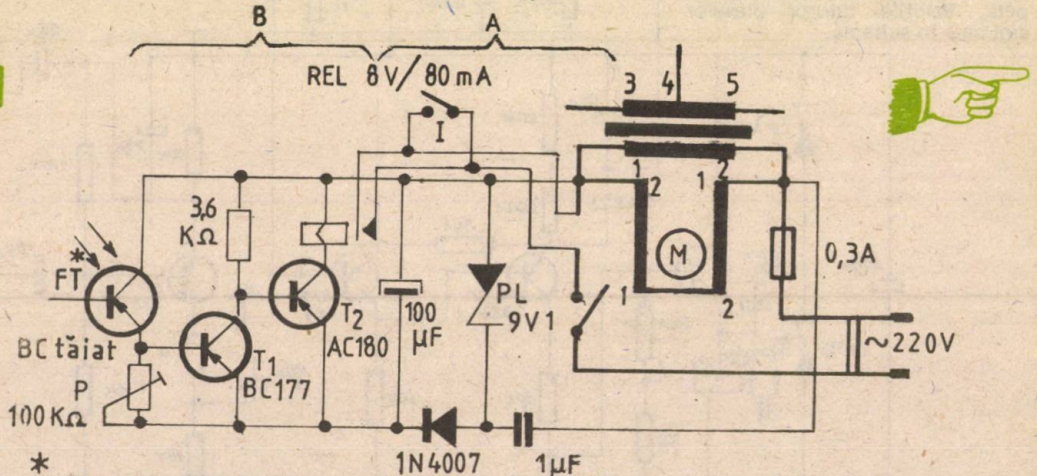
Ca sursă de fascicul luminos am folosit unul din cele două becuri de sub VU-metre și l-am montat cu ajutorul unei benzi de tablă de aluminiu pe primul ghidaj, dinaintea capului de ștergere. Tot pe aceeași bucată de tablă am montat și fototranzistorul (personal am folosit un BC177 cu capsula tăiată).

Pentru o acționare mai fermă, e bine ca banda să fie prevăzută la capătul porțiunii magnetice cu o fantă (se realizează prin ștergere cu diluant), deoarece porțiunea colorată de la capete prezintă o oarecare transparență. Întrerupătorul I va fi amplasat lîngă cel original al magnetofonului, pentru comoditate, și întregul montaj va fi izolat de

2



1



* Folosind ROL 31÷33 se inversează emițătorul și colectorul

COMANDĂ prin TENSIUNE, COMANDĂ prin CURENT

5

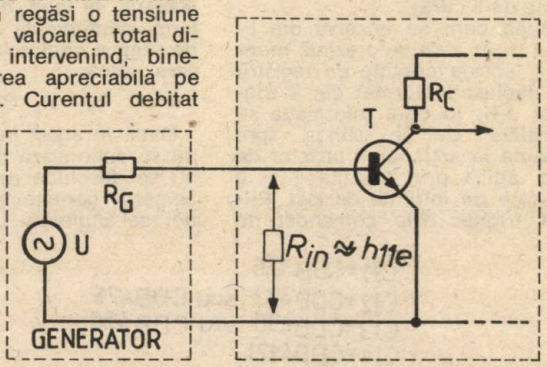
Fiz. A. MĂRCULESCU

În descrierea diverselor montaje electronice întîlnim adeseori, în special cu referire la tranzistoare, noțiunile de **comandă prin curent, comandă prin tensiune** sau **comandă adaptată.** În cele ce urmează vom analiza semnificația acestor noțiuni pe baza schemei particulare din figură, cu mențiunea că ele caracterizează în general cuplajele electrice dintre două circuite, blocuri, etaje ale aceluiași montaj etc.

Simplificat, figura ilustrează cuplajul dintre o sursă (generator) de tensiune alternativă U și un etaj de amplificare cu un tranzistor în conexiune EC. S-au mai specificat pe figură rezistența internă a generatorului, R_G, și rezistența de intrare a etajului de amplificare, R_i (aproximativ egală în acest caz cu parametrul h_{11e} al tranzistorului).

Sa presupunem întîi că R_G este foarte mică în comparație cu R_i. În această situație tensiunea alternativă la intrarea tranzistorului este practic egală cu tensiunea U debitată de generator (căderea pe R_G fiind neglijabilă). Se spune că tranzistorul este comandat prin/în tensiune, semnalul de comandă fiind practic egal cu tensiunea la bornele sursei în gol.

Dacă, din contră, R_G este foarte mare în comparație cu R_i, este evident că la intrarea tranzistorului vom regăsi o tensiune alternativă cu valoarea total diferită de U, intervenind, bineînțeles, căderea apreciabilă pe rezistența R_G. Curentul debitat de generator nu se modifică practic atunci cînd scurtcircuităm între ele bornele de intrare ale tranzistorului. Este ca și cum tranzistorul ar fi comandat de curentul maxim pe care îl poate furniza sursa, respectiv curentul de scurtcircuit al generatorului. Se spune în acest caz că tranzistorul este comandat prin/în curent.



Între aceste două cazuri extreme se plasează numeroasele situații practice posibile din punct de vedere al raportului dintre rezistențele (impedanțele) R_G și R_i. Un important caz particular îl reprezintă egalitatea celor două rezistențe, $R_G = R_i$, cînd (se poate ușor demonstra) transferul de energie de la generator la amplificator este maxim. Se spune că avem de-a face cu o comandă adaptată (sau cuplaj adaptat) în impedanță, situație preferabilă întotdeauna din punct de vedere al randamentului energetic. După cum se știe însă, în proiectarea circuitelor electronice criteriul transferului optim de energie (putere) nu este nici singurul și, adeseori, nici cel mai important, avîndu-se în vedere totodată simplificarea montajelor, din considerente de cost, fiabilitate, gabarit etc. Există, desigur, situații în care condiția adaptării de impedanță este critică și atunci constructorul amator trebuie să o respecte ca atare.

UMOR

șasiu, deoarece are 220 V la masă. Releul va fi fixat cu ajutorul unor garnituri de cauciuc pentru a nu vibra din cauza difuzorului. Piesele pot fi montate și „în aer", fiind puține la număr.

Amplasarea tuturor elementelor se vede în figura 2, unde am notat cu 1—întrerupător suplimentar (I); 2—fototranzistor; 3—bec de acționare a autostopului (24 V/50 mA); 4—releu; 5—montaj.

MOD DE FOLOSIRE

Se pornește magnetofonul din întrerupătorul lui original. Dacă becurile de control se aprind, înseamnă că I este închis. Dacă nu, se apasă întrerupătorul și întregul aparat intră în funcțiune. Se introduce banda normal în ghidaje; în acest moment, releul anclanșează. Se apasă din nou întrerupătorul I și becurile trebuie să rămînă aprinse. La o nouă cădere a luminii pe fototranzistor, contactele releului se eliberează și astfel se întrerupe funcționarea întregului aparat.

5

CIFRU

Ing. C. VLAHIȚĂ

Circuitul se poate utiliza pentru limitarea accesului în clădiri, încăperi sau diferite seifuri, ale căror sisteme de deschidere sînt blocate de miezul mobil al unui electromagnet.

Deschiderea acestora presupune cunoașterea și formarea corectă a unui cod (o succesiune de 5 cifre).

După cum se observă din figura 1, în care se prezintă montajul, acesta conține un registru de deplasare, format din 4 triggere J-K, în care informația se transferă de la stînga spre dreapta la acționarea tastelor de cod, adică prin tranziții 1 → 0 aplicate pe intrările de tact. Primul trigger este comandat de tasta K_1 pe intrarea S, acționare ce determină Q = 1. Avînd acum J = 1, triggerul $CI_{1\cdot2}$ își va schimba starea pe frontul negativ al unui impuls de tact, adică la acționarea tastei K_2. Procesul decurge similar și pentru triggerele $CI_{2\cdot1}$ și $CI_{2\cdot2}$.

Cînd ieșirea Q a ultimului trigger este „1", poarta NAND CI_3 își schimbă starea la ieșire, întrucît toate intrările sale se află la nivelul „1" logic. Această schimbare determină basculareа circuitului monostabil CI_4 (CDB4121). Nivelul „1" ce apare pe ieșirea Q a acestui circuit deschide tranzistorul T_1 și prin contactele releului se aplică tensiune la bobina electromagnetului și la contactul K_5.

Ieșirea $\overline{Q}$ = 0 de la monostabil aduce registrul de deplasare în starea „0", adică fiecare trigger va avea Q = 0.

În această nouă stare, monostabilul se află o perioadă de timp determinată de constanta de timp a circuitului de temporizare, adică:

$$T_0 = R_t \cdot C_t \cdot \ln 2$$

Dacă în acest interval de timp nu se acționează contactul K_5 și nu se deschide ușa, eventualele comenzi corecte efectuate anterior se anulează. Durata intervalului T_0 se alege în practică egală cu 3 — 6 secunde.

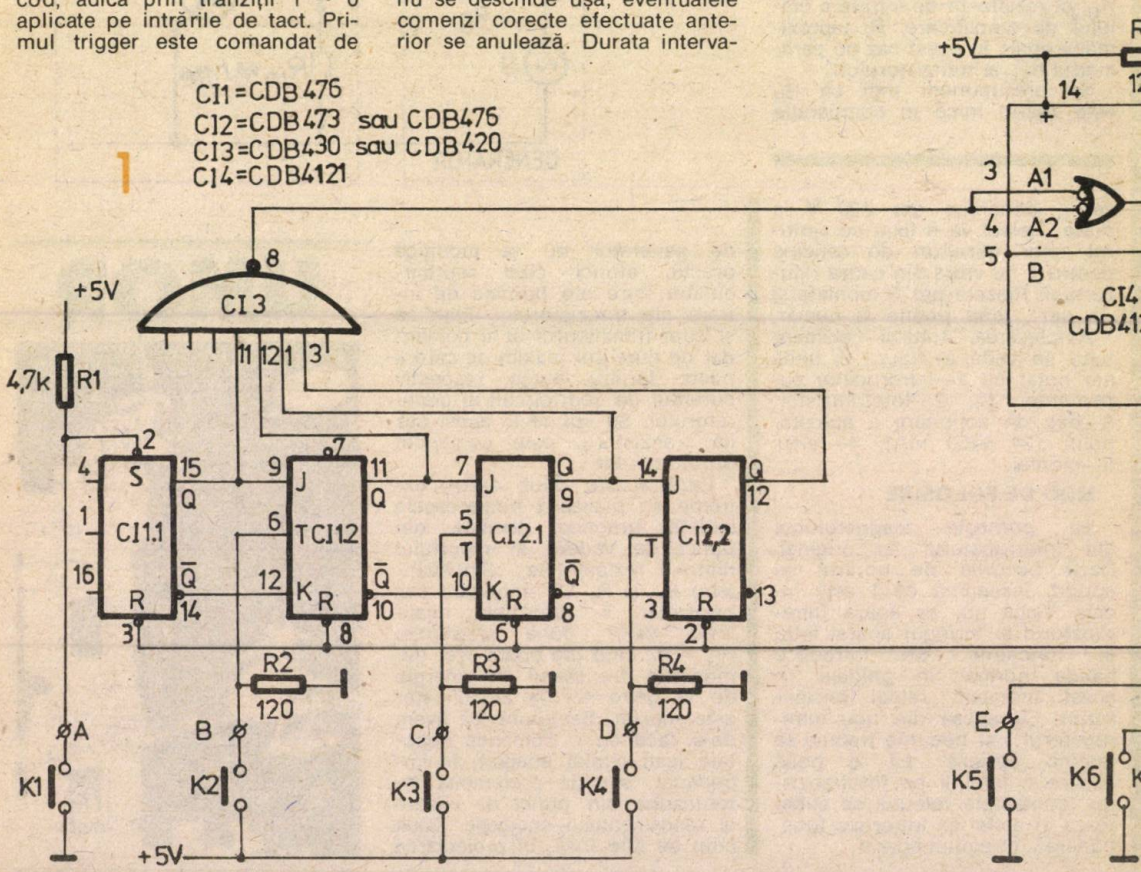
În concluzie, accesul este permis numai dacă se formează codul corect, adică dacă se apasă, în ordinea prestabilită, tastele K_1 — K_5.

Panoul pe care se află montate aceste taste conține și o serie de taste false, notate în schemă cu: K_6 — K_n. În figura 2 se reprezintă o modalitate de amplasare a acestor taste, pentru cazul n = 12.

Fiind conectate între masă și intrarea de RESET a triggerelor J—K, tastele false vor anula, în cazul că sînt acționate accidental, eventualele comenzi corecte, anterioare. Contactele K_1 — K_5 pot fi în orice poziție pe panoul cu taste. De exemplu: K_1 = 3; K_2 = 11; K_3 = 2; K_4 = 9 și K_5 = 1.

În acest caz, codul ce trebuie memorat și cu care se acționează corect circuitul este: 3 — 11 — 2 — 9 — 1.

Evident, aceste combinații se pot modifica la nevoie, schimbînd amplasarea tastelor K_1 — K_5 pe panou. Trebuie remarcat că se pot folosi și coduri în care aceeași cifră se repetă, dar în



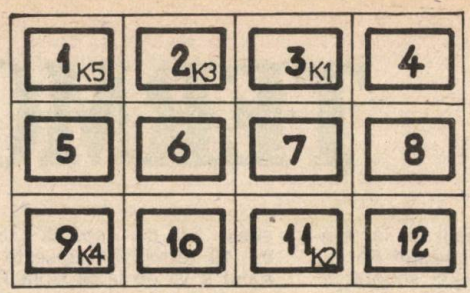
acest caz aceeași tastă va acționa asupra a două triggere diferite. Aceasta nu este valabil pentru primul trigger.

Schema se poate modifica de asemenea în sensul renunțării sau adăugării de noi triggere în registrul de deplasare. Codul va scădea ca număr de cifre sau se va lungi corespunzător. Consumul măsurat pe un montaj experimental a fost de 43—45 mA la 5 V.

Cea mai complicată problemă constă în realizarea panoului de taste și a părților mecanice aferente mecanismului de blocare.

Dacă în ceea ce privește ultimul, soluția se va alege în funcție de destinația concretă a cifrului electronic, pentru punctul de taste se pot da cîteva sugestii. Se pot utiliza microîntrerupătoare fabricate la Întreprinderea „Elecroaparataj" sau tastele de la un calculator de buzunar defect. Se poate, de asemenea, utiliza o tastatură construită după procedeul descris în articolul: „Tastatură mecanică extraplată" publicat în revista „Tehnium" nr. 9 din 1983. Membranele pocnitoare se pot înlocui cu lamele arcuite, dar în acest caz între folia de plastic colorat și cea de prespan se mai introduce încă o bucată de carton prespan cu 12 decupări circulare. Lamelele se află între cele două folii de carton prespan.

2



5

SEMNALIZATOR de REȚEA

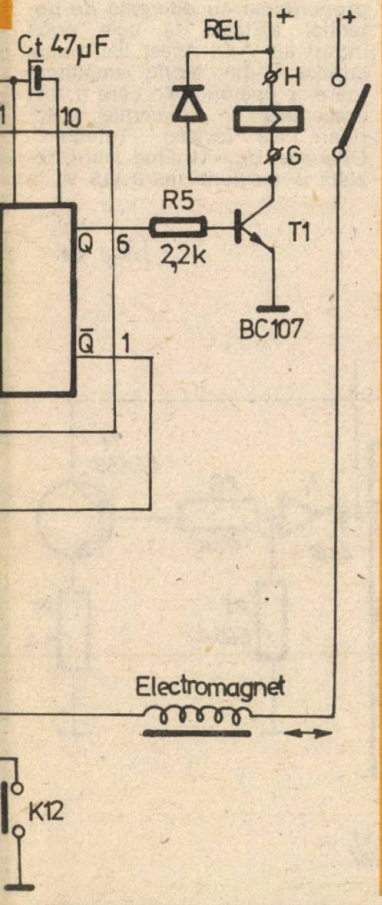
Ing. ADRIAN SÂRB

Schemele prezentate sînt destinate pentru indicarea locului de amplasare a comutatoarelor folosite la sistemul de iluminat al scărilor, subsolurilor, apartamentelor etc.

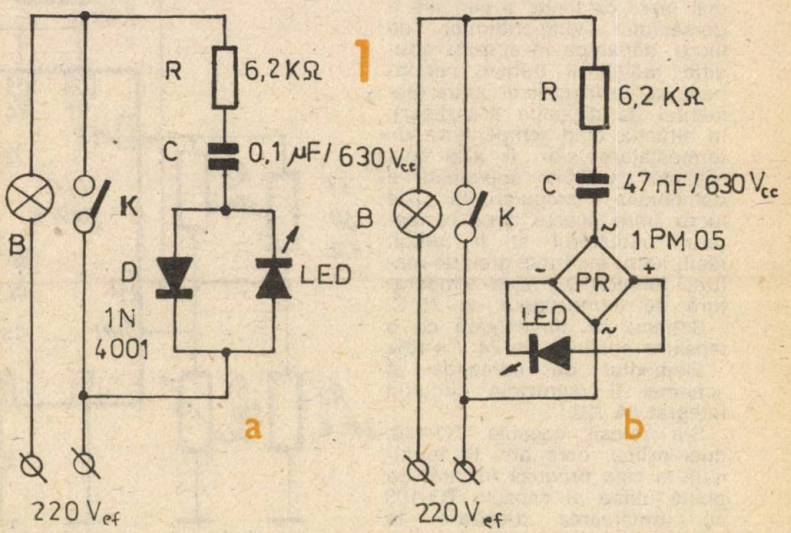
Elementul de semnalizare este un LED montat într-o gaură cu diametru corespunzător, practicată în capacul comutatorului. Celelalte elemente aferente schemei alese se introduc în doza comutatorului, după ce s-au efectuat conexiunile necesare. Terminalele elementelor trebuie izolate cu tub din PVC.

Considerînd comutatorul K deschis și varianta din figura 1a, prin circuit se stabilește un curent dependent de reactanța condensatorului și rezistență. Rolul rezistenței este de a limita valoarea de vîrf a curentului în momentul deschiderii comutatorului K. Dioda D asigură protejarea LED-ului pentru alternativa în care acesta ar fi blocat.

Varianta din figura 1b oferă posibilitatea de a obține prin LED o valoare medie a curentului dublă față de schema din figura 1a. Acest fapt permite utilizarea unui condensator cu valoarea capacității mai mică, deci cu gabarit mai redus.



1



5 TERMOSTAT

Ing. ŞERBAN NAICU

Schema este realizată în principal cu circuitul integrat βA 723 şi tranzistorul BD439 scopul urmărit şi realizat este acela de a menţine o temperatură cît mai constantă (cu precizia de 1°C, de exemplu) pe un radiator sau într-o incintă. Acest lucru este necesar pentru a menţine constantă temperatura unui dispozitiv sau echipament care are o comportare fluctuantă cu variaţia temperaturii mediului ambiant. De exemplu, menţinerea constantă a temperaturii unui cristal de cuarţ care piloteză un oscilator şi unde se cer stabilităţi ale frecvenţei de oscilaţie, în toată gama temperaturilor de lucru, foarte bune (sub $\pm 5.10^{-7}$ p.p.m.). Acest echipament reprezintă un oscilator termostatat şi acesta este şi scopul principal pentru care termostatul prezentat a fost proiectat.

Tranzistorul T_1 se află fixat pe un radiator, iar în imediata lui apropiere pe acelaşi radiator se află fixat şi termistorul R_T. Pe acelaşi radiator se află amplasat şi cristalul de cuarţ sau dispozitivul a cărui temperatură trebuie menţinută constantă în tot domeniul temperaturilor de lucru. S-a considerat o gamă a temperaturilor de lucru foarte extinsă (−40°C, +70°C), care acoperă practic orice aplicaţie. Temperatura la care va trebui termostatat radiatorul trebuie să fie mai mare ca limita superioară a domeniului temperaturilor de lucru, deoarece în această condiţie radiatorul trebuie permanent să fie încălzit de către elementul de disipaţie (tranzistor); în situaţia cînd temperatura de termostatare s-ar fi ales mai scăzută ca limita superioară a domeniului temperaturilor de lucru între aceste două temperaturi radiatorul ar fi trebuit răcit, lucru mult mai greu de realizat tehnic. S-a ales temperatura de termostatare de 75°C.

Schema se alimentează cu o tensiune continuă de 24 V ±10%.

Elementul de comandă al schemei îl reprezintă circuitul integrat βA 723.

S-a folosit capsula TO-116, dual-in-line, care are 14 terminale şi este produsă în ţară. Se poate utiliza şi capsula TO-100 cu următoarea precizare: la acest tip de capsulă (rotundă, cu 10 terminale) lipseşte terminalul 9 Vz de la capsula TO-116, terminal folosit în schema noastră. Dacă se studiază schema electrică a circuitului integrat βA 723 (prezentată în cataloagele I.P.R.S. şi I.C.C.E.), se observă că terminalul Vz este practic terminalul V_o, la care s-a înseriat o diodă Zener de 6,2 V (cu catodul montat la V_o).

Rezultă că se poate folosi fără probleme şi acest tip de capsulă, modificîndu-se în schimb valoarea diodei D_1 din schema noastră de la valoarea de 10 V la valoarea însumată a celor două diode Zener, şi anume la o valoare de 16 V (DZ 16).

Circuitul integrat βA 723 are în componenţa sa în principal un amplificator operaţional şi o sursă de tensiune stabilizată $V_{REF} = 7,15$ V.

Se poate observa pe schemă că tensiunea de referinţă este folosită la alimentarea divizoarelor rezistive R_1—R_2, R_3—R_T, care au rolul de a polariza intrările inversoare şi neinversoare ale amplificatorului operaţional.

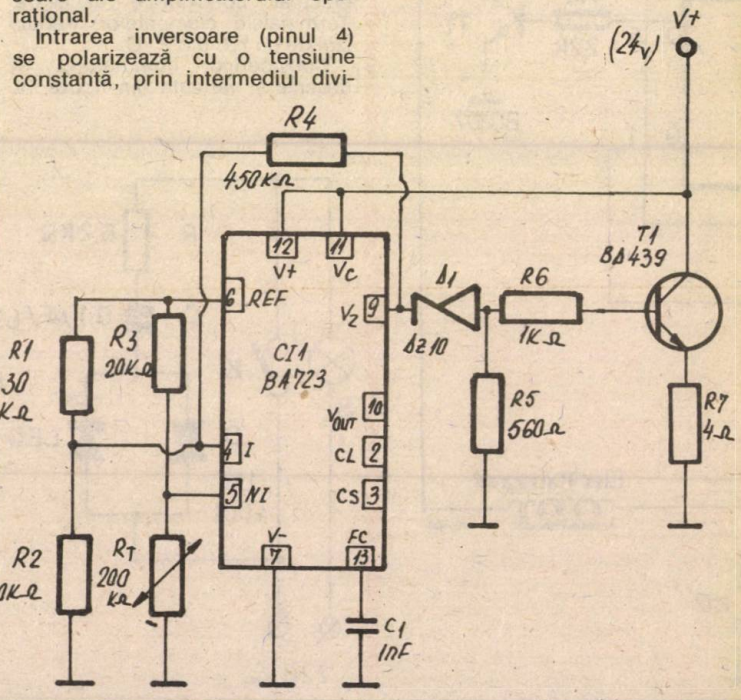
Intrarea inversoare (pinul 4) se polarizează cu o tensiune constantă, prin intermediul divizorului rezistiv R_1—R_2, de valori egale, care este jumătate din valoarea tensiunii de referinţă.

Deci $U_I = \frac{7,15\ V}{2} = 3,57$ V.

Intrarea neinversoare a operaţionalului (pinul 5) se polarizează prin intermediul divizorului R_3—R_T. R_T este un termistor cu coeficient de temperatură negativ (NTC) de tip subminiatură, în sticlă (produs la I.F.T.M.-Măgurele).

Cînd termistorul este rece, el va prezenta o rezistenţă mare. Termistorul ales de noi prezintă la temperatura ambiantă o rezistenţă de 200 kΩ. Deci la pornire se va aplica o tensiune mare la pinul 5 al C.I. (căderea de tensiune de pe termistor) de 6,18 V.

Potenţialul la pinul 9 (Vz) este proporţional cu diferenţa de potenţial aplicată la cele două intrări ale A.O. Acest lucru este cunoscut din teoria amplificatoarelor operaţionale care a fost prezentată în numerele anterioare ale revistei „Tehnium". Diferenţa U_{NI}—U_I fiind mare, rezultă o tensiune mare (15 V) la



pinul 9 (Vz). Această tensiune se aplică prin intermediul grupului D_1, R_6 la baza tranzistorului T_1, pe care o polarizează, determinînd „deschiderea" tranzistorului. În acest fel curentul de colector al tranzistorului începe să crească, determinînd încălzirea tranzistorului. Implicit se încălzește și termistorul care este montat pe același radiator cu tranzistorul (prin fixare într-un tubuleț metalic cositorit pe radiator). Prin încălzire rezistența termistorului începe să scadă. Astfel scade și tensiunea care „cade" pe el, deci tensiunea aplicată la pinul 5 al C.I. Se micșorează și diferența U_{NI}—U_I, deci și tensiunea la ieșirea A.O. (pinul 9 Vz). Rezultă că tensiunea care se va aplica în baza tranzistorului va începe să scadă, deci tranzistorul va începe să se blocheze, începînd să nu mai disipe căldură, deci să se răcească. Termistorul se va răci și el, rezistența lui va crește din nou și deci tot ciclul se va relua. Termistorul se va alege astfel ca la temperatura de termostatare a radiatorului aleasă (la noi 75°C) rezistența lui să fie egală cu R_3. În această situație, U_I devine egală cu U_{NI} deci Vz este minimă.

Valorile R_1, R_2, R_3, R_T nu sînt critice, ele pot fi alese în funcție de ce dispunem, cu respectarea condiției de mai sus și ținînd cont că I_{REF} = 15 mA, deci curentul prin ramura R_1, R_2, însumat cu cel prin ramura R_3, R_T, nu trebuie să depășească valoarea prezentată, în caz contrar integratul se distruge.

Dioda D_1 este un Zener de 10 V și are rolul de a limita tensiunea care ajunge în baza tranzistorului, evitînd astfel creșterea curentului prin joncțiunea B-E peste limita de străpungere.

Curentul de pornire (practic curentul de colector al tranzistorului) este de maximum 600 mA și este limitat prin intermediul lui R_7. Dată fiind puterea pe care o disipă această rezistență, se recomandă a fi realizată prin bobinare pe corpul altei rezistențe (cu nichelină, de exemplu).

Rezistența R_4 (reacție ieșire-intrare inversoare) contribuie la funcționarea circuitului integrat ca amplificator operațional. În lipsa ei, acesta ar fi lucrat ca un comparator, prezentînd inconvenientul că termostatul ar fi consumat curent în trepte (cu impulsuri) cu dezavantajele inerente pentru sursa de alimentare.

Puterea la pornire este de circa 15 W, iar în regim de funcționare de 1,5 W.

Timpul de intrare în regim este de circa un minut.

Se recomandă ca rezistențele (excepție R_7 bobinată) să fie de tip RPM, putere disipată 0,25 W, ar C_1 multistrat (produs de I.C.E.P.-Curtea de Argeș).

Respectînd aceste cerințe și recurgînd la o proiectare îngrijită a cablajului, montajul prezentat mai sus încape pe o plăcuță de sticlotextolit cu o suprafață foarte mică (20 x 25 mm).

ATENȚIE, ÎNCEPĂTORI!

În cataloagele firmelor producătoare, performanțele unui anumit dispozitiv semiconductor sînt de obicei exprimate prin mărimi (valori) caracteristice și prin valori maxime admisibile (maximum ratings).

Mărimile caracteristice sînt acelea care pot fi măsurate, folosind instrumente și circuite adecvate, ele oferind informații despre performanțele dispozitivului în condiții date de lucru (de exemplu, pentru o tensiune de polarizare dată, pentru un anumit curent de colector etc.). După caz, ele sînt precizate prin valorile tipice și/sau valorile extreme garantate (minime, maxime). Uneori valorile caracteristice nu sînt exprimate numeric, ci sub forma unor curbe, punîndu-se în evidență plaja maximă de variație datorată împrăștierii din fabricație.

Valorile maxime admisibile desemnează valorile unor parametri de lucru care nu pot fi depășite fără riscul de a deteriora dispozitivul. În urmărirea acestui imperativ trebuie avute în vedere toleranțele (abaterile) celorlalte componente din circuit, precum și fluctuațiile posibile ale tensiunii de alimentare.

În legătură cu valorile maxime admisibile se impune o observație importantă: nici una dintre aceste valori nu poate fi depășită (fără risc), chiar dacă dispozitivul funcționează cu toți ceilalți parametri lejer sub limitele maxime admisibile.

Atenție, deci, începători! Un tranzistor poate fi ușor „ars" fără a depăși nici curentul său de colector maxim admisibil, nici tensiunile maxime de polarizare. O greșeală frecventă în astfel de cazuri este neglijarea valorii maxime admisibile a puterii de disipație.

UMOR

SELECTOR LOGIC

VIOREL VANEA

În varianta prezentată, montajul răspunde favorabil după formarea, pe o tastatură zecimală, a 16 cifre de la 0 la 9, ceea ce înseamnă că numărul variantelor ce se pot forma se ridică la 10^{16}. Chiar dacă se exclud cazurile particulare, rămîn totuși la îndemînă un număr destul de mare de variante.

Codul se stabilește prin contactele ce se realizează între liniile și coloanele selectorului: cifra de rang K poate lua valoarea m ($K \leq 15$, $m \leq 9$, $K\ m \in N$). Se observă că cifre de ranguri diferite pot lua aceeași valoare, pe cînd nu există două valori care să corespundă simultan aceluiași rang. Mai explicit, funcția definită pe mulțimea rangurilor, cu valori în mulțimea primelor 10 numere naturale, este o surjecție, pe cînd funcția definită pe mulțimea (0, 1, 2...9), cu valori în mulțimea rangurilor, este injectivă. Acest deziderat se rezolvă prin însăși arhitectura selectorului.

Schema (fig. 1) are ca element de bază etajul numărător pînă la 16, realizat cu CDB 493 și SN 74154.

Inițial, nivelul logic „L" se află la rangul 0 (ieșirea nr. 1). Dacă se acționează pentru un timp tasta corespunzătoare acestui rang (în cazul de față, 3), la ieșirea inversorului I_1 va apărea semnalul negat, adică un impuls „H" care duce la schimbarea stării întregului bloc numărător.

Astfel, următorul rang cade în „L" și așteaptă formarea cifrei corespunzătoare ș.a.m.d.

O dată cu formarea celei de-a 16-a cifre, poarta NAND realizată cu D17, D18, R2 declanșează monostabilul CI4 printr-un

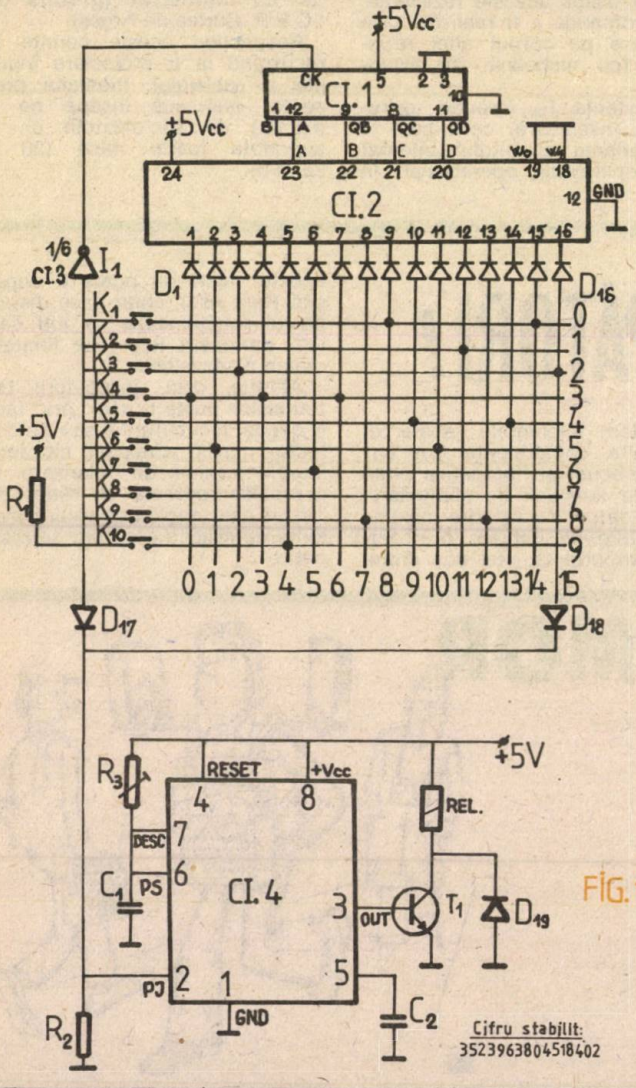


FIG. 1

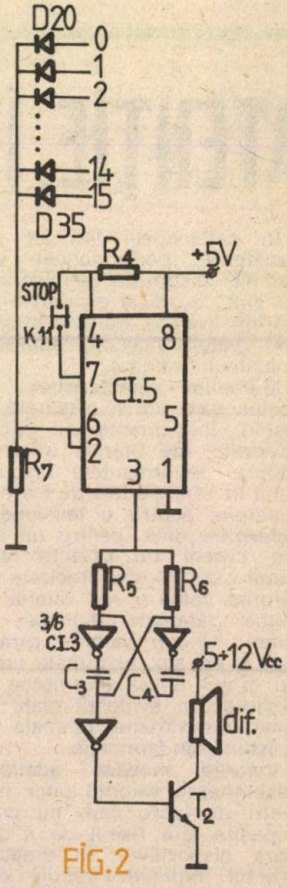


FIG. 2

impuls negativ avînd lățimea egală cu timpul de apăsare al tastei. Dacă impulsul este mai îngust decît 1,1 x R3 x C1, releul este alimentat numai pentru temporizarea fixată; cînd comanda se prelungește peste această valoare, timpul de alimentare a releului depinde numai de durata acționării tastei.

Cînd curentul absorbit de releu permite (maximum 200 mA), se poate renunța la tranzistorul T1, releul conectîndu-se între ieșirea integratului și +Vcc.

De remarcat că în acest caz se poate utiliza un releu ce funcționează la orice tensiune cuprinsă între 4,5 și 18 V, nemaifiind necesară corelarea cu V_{BEO} a tranzistorului utilizat (la seria $|V_{BEO}| = 5$ V).

Pentru R3 = 5,9 MΩ și C1 = 0,47 μF, temporizarea este de 3 s. În acest interval, elementul de execuție (o broască electromagnetică, de exemplu) permite deschiderea unei uși, pornirea unui motor etc.

Pentru cazul în care persoane neavizate încearcă să „înșele" montajul, s-a prevăzut (fig. 2) un dispozitiv de alarmă. Cînd unui anumit rang nu i se dă valoarea corectă, dioda corespunzătoare din circuitul de alarmă intră în conducție și face ca pinii „PRAG SUS" și „PRAG JOS" ai integratului CI 5 să urce în nivel „H", ceea ce duce la comutarea terminalului „RESET" (4). Prin urmare, tensiunea la ieșire scade în „L" și rămîne astfel, deoarece comanda „RESET" are prioritate față de „PRAG SUS" și „PRAG JOS". Oprirea alarmei se face printr-o scurtă întrerupere a contactului, K11.

INDICAȚII DE MONTAJ

Se va acorda atenție deosebită decuplării integratelor CI 1 — CI 4 (CI 5 are doar rol de comutator) cu condensatoare de calitate, amplasate cît mai aproape de pinii de alimentare.

Pentru alimentare, se va utiliza unul din numeroasele redresoare stabilizate publicate în „Tehnium".

În lipsa unui SN 74154, se poate utiliza un CDB 442 E, cu diminuarea corespunzătoare a numărului de variante (în acest caz 10^{10}).

Tastatura poate fi de orice tip, dar mai indicate sînt cele pentru telefoane (seria KAT).

Matricea de selecție poate fi o matrice programatoare ED 21, împreună cu 16 fișe cu diode, ED 22 (diode 1 N 4005).

În loc de βE 555 N/H se pot folosi și βE 555 E, ținîndu-se cont de noua configurație a terminalelor.

Pentru facilitarea memorării, se va folosi un cifru format din două numere telefonice complete (cu prefixe) preferat din localități diferite.

LISTA DE PIESE

CI 5 = βE555H, βE555N; T_2 = BD 139 ÷ 239; C3 = C4 = 0,1 ÷ 0,47 μF; D 20 ÷ D 35 = 1N4151; R_4 = 10 kΩ; $R_5 = R_6$ = 820 Ω — 2 kΩ; DIF: 4 ÷ 16 Ω/3 W; R_7 = 2 ÷ 2,5 kΩ (MAX. 5 kΩ); CI1 = CDB 4936; CI2 = SN 74154; CI3 = CDB 404E; CI4 = βE555H (TO 99), βE555N (MP 48); T_1 = BD 139 ÷ BD 239; $D_1 ... D_{19}$ = 1N4151; R_1 = 820 Ω; R_2 = 1 kΩ; R_3 = 5,9 MΩ; C_1 = 0,47 μF; C_2 = 10 ÷ 15 nF.

IPRS BANEASA

CONDENSATOARE ELECTROLITICE

Eg 7400

TIP/TYPE	C_n (μF)	U_n (V_{cc})	I_o (mA)	Z (ohmi)
EG 74.75	2200	16	1200	0,10
EG 74.77	3300	16	1800	0,10
EG 74.78	4700	16	2500	0,10
EG 74.80	6800	16	3000	0,10
EG 74.80	1000	16	3500	0,10
EG 74.73	680	25	680	0,26
EG 74.74	1000	25	950	0,18
EG 74.76	1500	25	1300	0,12
EG 74.77	2200	25	1600	0,10
EG 74.80	3300	25	2200	0,10
EG 74.80	4700	25	2500	0,10
EG 74.80	6800	25	3000	0,10
EG 74.73	470	40	680	0,32
EG 74.74	680	40	930	0,22
EG 74.76	1000	40	1260	0,15
EG 74.78	1500	40	1700	0,10
EG 74.80	2200	40	2130	0,10
EG 74.80	3300	40	2300	0,10

(CONTINUARE ÎN PAG. 82)

5

memorator

IPRS BANEASA

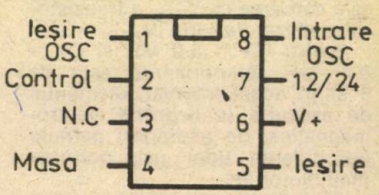
TBA 315 N este un circuit integrat proiectat pentru a realiza impulsuri dreptunghiulare cu reglarea duratei impulsurilor și a factorului de umplere printr-un grup RC, exterior circuitului.

Circuitul conține un etaj stabilizator de tensiune, un etaj comparator oscilator și un etaj final de putere.

Acest circuit permite excitarea ciclică a unui releu.

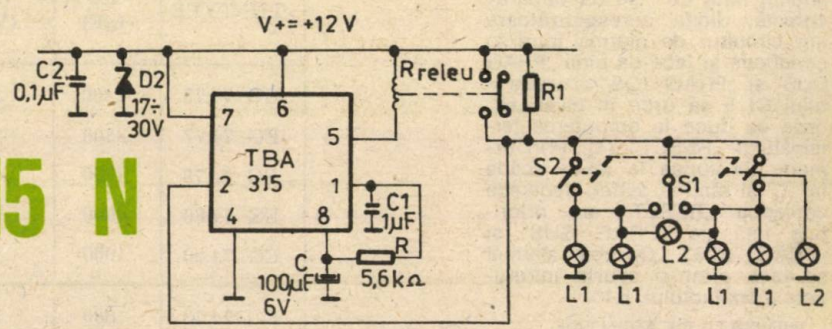
Este recomandat a fi utilizat în aplicații cum ar fi: semnalizarea direcției autovehiculelor, temporizare reglabilă a ștergătorului de parbriz și multivibratoare de putere.

Pentru V + = 12 V terminalul 11 (12/24) se conectează la V +, iar pentru 24 V terminalul acesta este neconectat.



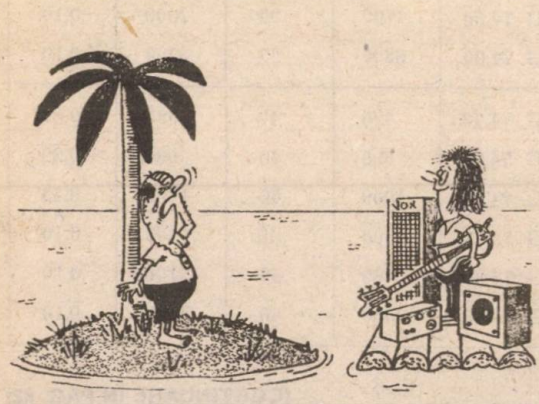
TBA 315 N

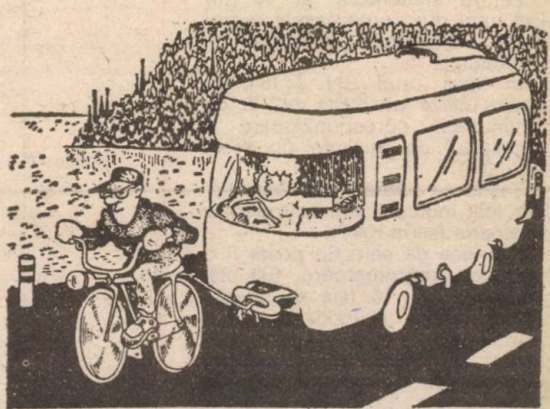
TBA 315 N



UMOR

Semnalizarea direcției la autovehicule.





STRUCTURA și FUNCȚIONAREA MEMORIILOR utilizate în microprocesoare

Prof. MIHAI CORUȚIU

Dispozitivul capabil să rețină o informație care i-a fost încredințată poartă numele de memorie. Pentru tipul de memorie care ne interesează, această informație este formată dintr-un ansamblu de niveluri logice.

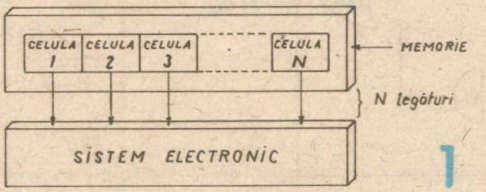
Cea mai simplă memorie pe care ne-o putem imagina — numită celulă — este capabilă să rețină un singur nivel logic, fie 0, fie 1.

Reuniunea a N celule formează o memorie de capacitate N biți. Bitul este unitatea de informație elementară susceptibilă de a fi memorizată printr-un nivel logic egal fie cu 0, fie cu 1.

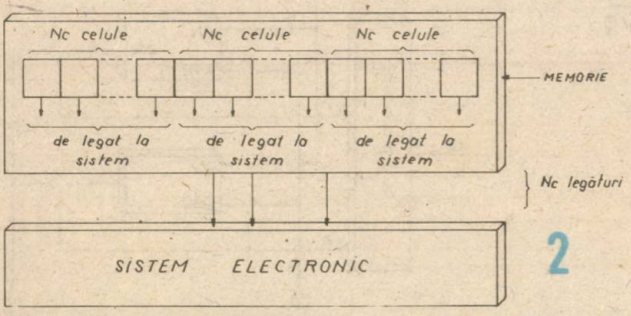
Trebuie menționat că informațiile conținute într-o memorie sînt destinate să comande un sistem electronic. În consecință, acest sistem trebuie să aibă acces la fiecare din informațiile elementare memorizate în celule. Acest acces poate fi materializat printr-un ansamblu de legături electrice între sistemul electronic și fiecare din ieșirile celulelor **(fig. 1)**. Soluția menționată este practic inadmisibilă deoarece necesită un număr foarte mare de legături: să ne imaginăm cazul în care N = 1 024!

Dificultatea apărută poate fi înlăturată în situația în care vom transmite sistemului electronic, în același timp, nu toate informațiile conținute în memorie, ci numai un anumit număr N_C, tipic egal cu 1, 4 sau 8. Cu alte cuvinte, trebuie ca, folosind un procedeu oarecare, să realizăm legătura cu sistemul electronic a numai N_C celule **(fig. 2)**.

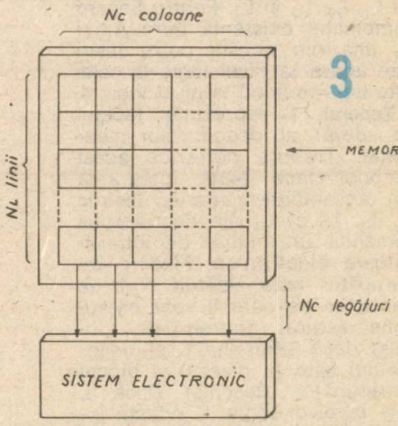
Pentru aceasta vom dispune celulele care formează memoria în linii și coloane. Fie N_L numărul de linii și N_C numărul de coloane **(fig. 3)**. Această organizare rezultă din necesitatea de a limita la un număr acceptabil legăturile memoriei cu sistemul electronic care trebuie comandat. În același timp, această organizare prezintă dezavantajul de a realiza un acces secvențial al sistemului electronic la me-



1



2



3

morie. Cu alte cuvinte, datele (informațiile binare) conținute în toate liniile nu pot fi furnizate sistemului electronic decît succesiv în timp.

O alta consecință importantă a acestei organizări este obligația de a dispune de semnale de comandă pentru memorie, destinate a selecționa linia ale cărei ieșiri trebuie legate la sistemul electronic care trebuie comandat.

Deoarece, la un moment dat, numai o singură linie este selectată, este suficient să dispunem de un număr p de semnale de selecție astfel încît $2^p \geq N_L$. De exemplu, 8 semnale sînt necesare și suficiente pentru selectarea unei linii dintr-un total de 256.

Fiecărei linii i se asociază un număr total care este numit adresă. Ansamblul format din cele p fire de selecție care suportă material adresa transmisă memoriei se numește magistrală de adrese.

Pentru a înțelege cum poate un „cuvînt" binar de p biți, reprezentînd adresa unei linii, să selecteze o linie din cele 2^p existente, ne vom referi la **figura 4**, care reprezintă o memorie cu capacitatea de 8 biți organizată în $N_L = 4$ linii și $N_C = 2$ coloane. Conform celor arătate mai înainte, doar două semnale sînt necesare și suficiente pentru a selecta o linie din cele patru; aceste două semnale sînt transmise prin firele notate A_1 și A_0, care formează magistrala de adrese. A_1 și A_0 sînt intrările unui decodor cu $2^2 = 4$ ieșiri, notate cu L_1, L_2, L_3 și L_4. Pentru fiecare combinație existentă între A_1 și A_0 una din aceste patru ieșiri este adusă la nivel logic 0, celelalte trei rămînînd la nivel logic 1.

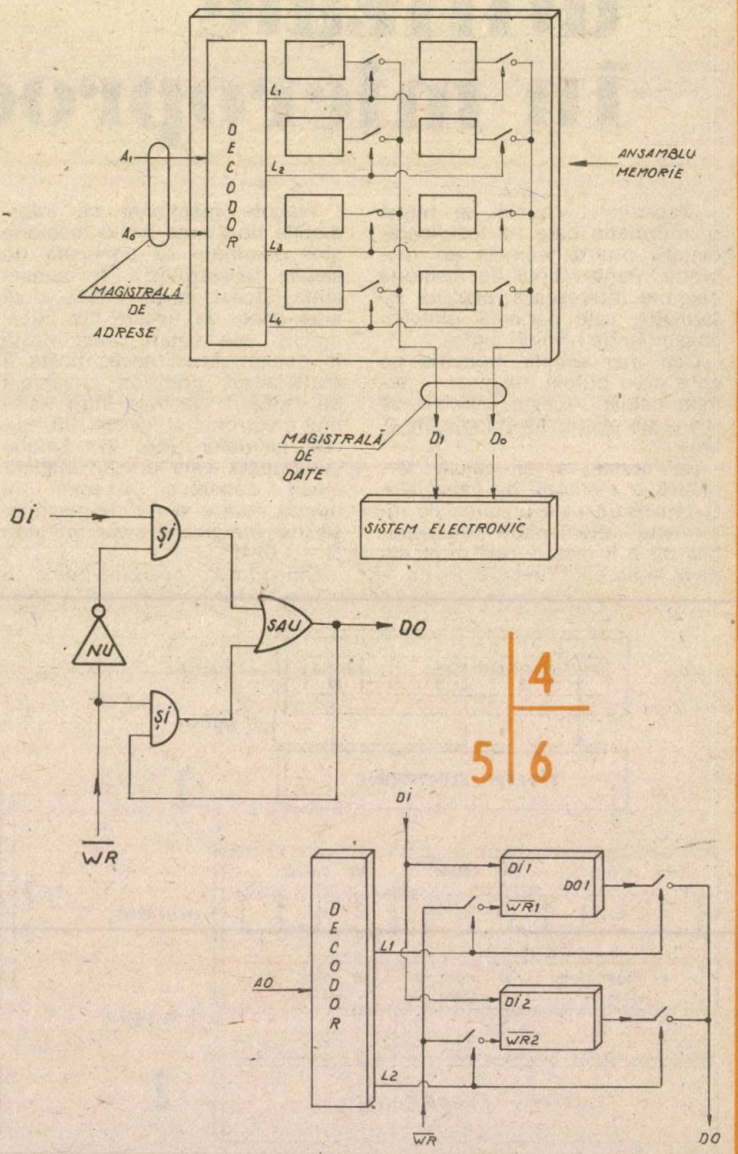
Tabelul 1 reprezintă tabelul de adevăr al decodorului menționat. Trebuie notat că acest decodor face parte integrantă din ansamblul-memorie. Ieșirile L_1, L_2, L_3 și L_4 ale decodorului comandă un număr de întrerupătoare electronice. Fiecare întrerupător este alcătuit dintr-un tranzistor pe care îl vom presupune saturat (întrerupător închis) dacă semnalul L al decodorului este la nivel 0 și blocat (întrerupător deschis) dacă L este la nivel logic 1. Aceste întrerupătoare au rolul de a lega ieșirile celulelor din linia adresată la sistemul electronic care trebuie comandat. Se spune că informațiile conținute în linia vizată sînt citite de către sistemul electronic. Ansamblul format din cele N_C fire de transmisie prin care circulă aceste informații, notate cu D_1 și D_0 în figura 4, formează magistrala de date.

Trebuie remarcat că întrerupătoarele menționate au două funcții importante:

— să selecteze o linie și numai una din cele N_L existente, al cărei conținut este destinat să fie citit de către sistemul electronic;

— să permită informațiilor elementare, conținute în fiecare din celulele ce compun o coloană, să circule pe aceeași cale de transmisie. Dacă această condiție nu este îndeplinită, atunci ieșirile mai multor celule vor fi legate între ele, adică se va produce un scurtcircuit.

Utilizarea întrerupătoarelor conduce la adoptarea unui termen nou, acela de logică 3-stări (Tri-State-Logic). Iată ce înseamnă aceasta. Presupunem că un observator este situat pe unul din firele magistralei de date, de exemplu pe firul notat cu D_1, și însărcinat să determine care este starea logică a ieșirii unei celule, de exemplu, cea corespunzătoare primei linii. Dacă



întrerupătorul respectiv este închis, răspunsul său va fi că sînt două stări logice posibile: 0 sau 1. Dacă întrerupătorul respectiv este deschis, răspunsul său va fi că există o a treia stare pentru care ieșirea celulei nu este conectată; se spune, în acest caz, că celula este în starea de „înaltă impedanță" (H.I.).

Utilizarea logicii 3-stări este extrem de avantajoasă pentru că ea simplifică considerabil interconectarea componentelor. De exemplu, pentru o memorie cu o capacitate de 1 kbit (unde 1 kbit = 1 024 biți), organizată în 128 linii și 8 coloane, 7 fire de legătură pentru adrese și 8 fire de legătură pentru date sînt suficiente pentru conectarea ansamblului-memorie la sistemul electronic care trebuie comandat. Pe de altă parte, lectura unei date nu se face instantaneu deoarece între momentul în care o adresă va fi transmisă pe magistrala de adrese și momentul în care linia adresată va fi disponibilă pe magistrala de date va trece un timp, numit timp de acces.

Timpul de acces depinde de tehnologia folosită; el este de cîteva sute de nanosecunde în tehnologia MOS și de cîteva zeci de nanosecunde în tehnologia TTL. Totuși memoriile realizate în tehnologia MOS sînt utilizate în mod frecvent pentru că ele permit fabricarea componentelor cu o mare densitate de integrare.

MEMORIILE RAM

Denumirea RAM provine din limba engleză: „Random Access Memory", ceea ce înseamnă „memorie cu acces aleator". Deoarece această memorie poate fi supusă unei succesiuni de scrieri și lecturi, rezultă că denumirea ei nu ar fi cea mai indicată.

Structura unei celule este cea a unui registru „latch" de capacitate 1 bit **(fig. 5)**. În această figură, DI este o dată care trebuie scrisă, iar DO este o dată care trebuie citită:

— dacă $\overline{WR} = 0$, atunci DO = DI; se spune că registrul este „transparent" pentru data DI prezentă la intrarea sa: data DI este scrisă în celulă;

— dacă $\overline{WR} = 1$, atunci DO este insensibil la eventualele variații ale lui DI. Pe de altă parte, starea logică a lui DO este cea imediat precedentă tranziției 0 → 1 a lui $\overline{WR}$: data DI este memorizată în celulă pentru frontul crescător al semnalului $\overline{WR}$. Semnalul $\overline{WR}$ (în limba engleză to write = a scrie) este semnalul de scris și este activ pentru nivelul logic zero; de aceea el este supraliniat.

Figura 6 arată în ce mod o dată prezentă în magistrala de intrare a datelor DI este scrisă numai în linia adresată. Pentru mai multă claritate a fost reprezentată o memorie cu capacitatea de 2 biți (două linii și o coloană). Calea de transmisie prin care circulă semnalul de scris $\overline{WR}$ este legată la intrările de comandă $\overline{WR1}$ și $\overline{WR2}$ ale celulelor prin intermediul întrerupătoarelor de comandă existente la ieșirile L_1 și L_2 ale decodorului. În consecință, numai celula sau, în general, numai linia adresată este scrisă. Se poate evidenția că data DI este prezentă și în magistrala de ieșire a datelor DO.

Fie o memorie RAM de capacitate 1 kbit organizată în 128 linii și 8 coloane. În acest caz, adresajul necesită 7 fire de legătură, intrarea de date 8 fire, iar ieșirea de date încă 8 fire. Dacă ținem seama de alimentarea montajului și de semnalul de scriere, vom totaliza 26 fire de legătură, ceea ce înseamnă foarte mult.

Pentru a reduce acest număr la o valoare acceptabilă, trebuie să transmitem datele de intrare și datele de ieșire pe o magistrală comună. Se poate face aceasta legînd DI la DO prin intermediul unor întrerupătoare comandate C_1 și C_2 **(fig. 7)**, așa încît:

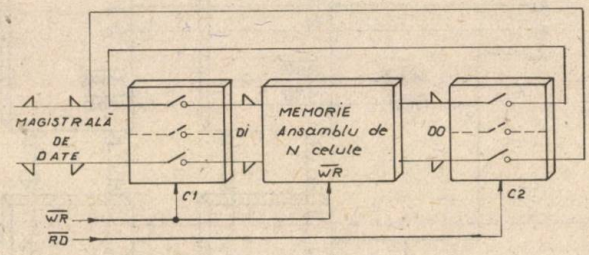
— dacă $C_1 = 0$ și $C_2 = 1$, atunci DI este același cu data prezentă în magistrala de date. Se spune că memoria este în poziția „scriere". Data respectivă este efectiv scrisă dacă $\overline{WR} = 0$;

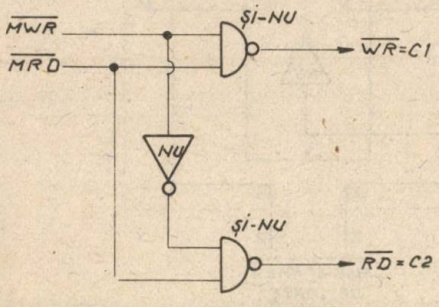
— dacă $C_1 = 1$ și $C_2 = 0$, intrarea DI a memoriei este întreruptă de magistrala de date, aceasta suportînd DO, cu alte cuvinte, conținutul memoriei corespunzător liniei adresate. Se spune că memoria este în poziția „lectură".

Rezultă că C_1 și C_2 sînt semnalele de scriere și de lectură (citire). Ca urmare, putem să legăm $\overline{WR}$ la C_1 și să punem $C_2 = \overline{RD}$ (în limba engleză: to read = a citi).

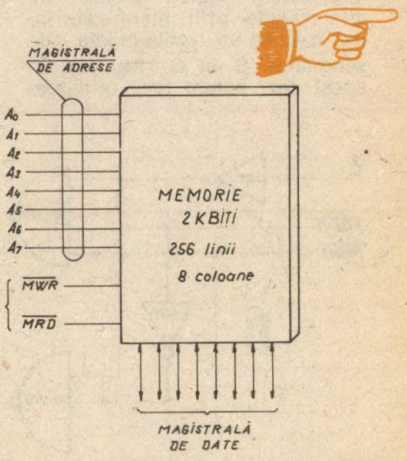
Remarcăm:

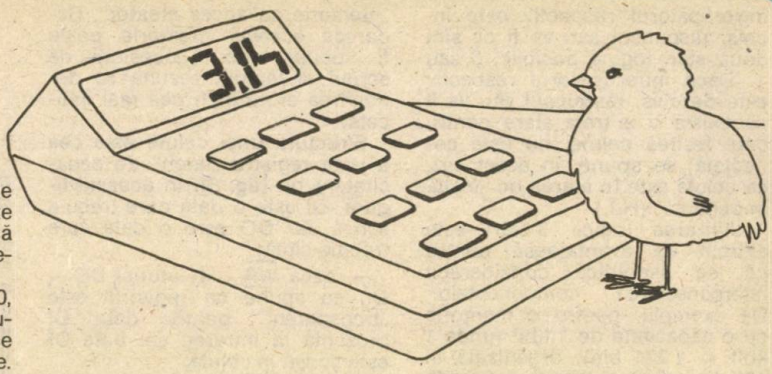
1. Dacă $\overline{RD} = 1$ și $\overline{WR} = 1$,





7
8|9





atunci memoria este izolată de magistrala de date. Nu se poate nici citi, nici scrie. Vom spune că ea este în starea de „înaltă impedanță".

2. Dacă $\overline{RD} = 0$ și $\overline{WR} = 0$, atunci magistrala de date suportă constant DI și DO, ceea ce interzice scrierea în memorie. Această combinație trebuie interzisă, ceea ce se poate realiza cu ajutorul montajului arătat în **figura 8.** Aici $\overline{MWR}$ (Memory Write) și $\overline{MRD}$ (Memory Read) sînt semnalele de control ale memoriei și sînt astfel încît:

— dacă $\overline{MWR} = 0$ și $\overline{MRD} = 1$, atunci memoria este în poziția „scriere";

— dacă $\overline{MWR} = 1$ și $\overline{MRD} = 0$, atunci memoria este în poziția „citire";

— dacă $\overline{MWR} = 1$ și $\overline{MRD} = 1$ sau $\overline{MWR} = 0$ și $\overline{MRD} = 0$, atunci memoria este în starea „înaltă impedanță".

Să considerăm o memorie de 2 kbiți organizată în 256 linii și 8 coloane **(fig. 9).** Magistrala de adrese este alcătuită din 8 fire notate de la A_0 la A_7, iar magistrala de date, care transmite informația de citit sau de scris, este constituită tot din 8 fire de legătură. Lectura și scrierea sînt controlate de semnalele $\overline{MWR}$ și $\overline{MRD}$. O astfel de memorie poate fi obținută din alte două avînd fiecare capacitatea de 1 kbit, organizate în 128 linii și 8 coloane. Soluția rezultă în urma cuplajului magistralelor de date pentru cele două memorii prin intermediul logicii 3-stări, cu alte cuvinte, prin intermediul întrerupătoarelor comandate de semnalele C și $\overline{C}$ **(fig. 10).** În acest caz, putem face următoarele afirmații:

— dacă $\overline{C} = 0$, memoria I este singura selecționată și, în consecință, numai ea este legată la magistrala de date. Bitul A_7 existent în magistrala de adrese, fiind legat la $\overline{C}$, determină ca memoria I să fie selecționată cînd adresa ($A_7A_6A_5A_4A_3A_2A_1A_0$) variază de la (00000000) la (01111111). În notația hexazecimală spunem că memoria I rezidă de la adresa 00 la 7F. Conversia binar-hexazecimal este arătată în **tabelul 2.**

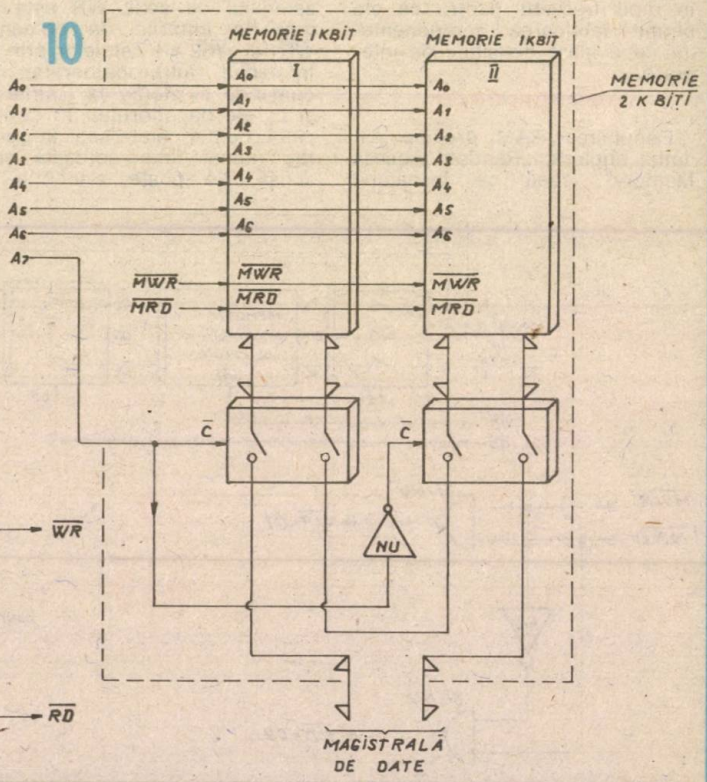
În situația în care $\overline{C} = 1$, adresa variază de la (10000000) la (11111111) și memoria II este singura legată la magistrala de adrese. Memoria II rezidă de la adresele 80 la FF.

Rezultă că semnalele de comandă C și $\overline{C}$ sînt semnale de selectare pentru ansamblul-memorie. De obicei, aceste semnale sînt notate $\overline{CS}$ sau CS (în limba engleză: „Chip Select").

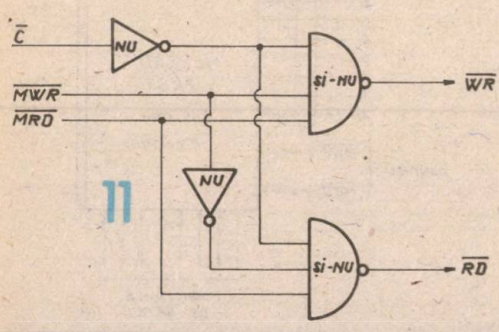
În realitate, dacă dorim să realizăm o memorie de capacitate 2 kbiți, avînd la dispoziție două memorii de 1 kbit fiecare, nu este neapărat necesar să folosim întrerupătoarele din figura 10. Pentru a înțelege aceasta, să ne fixăm atenția, de exemplu, asupra memoriei I:

— dacă $\overline{C} = 1$, memoria I trebuie plasată în starea de „înaltă impedanță", ceea ce se poate realiza foarte ușor impunînd ca în-

10



11



trările de comandă $\overline{WR}$ și $\overline{RD}$ din figura 7 să fie la un nivel logic 1;

— dacă $\overline{C} = 0$, memoria I trebuie selecționată, ceea ce impune ca intrările de comandă $\overline{WR}$ și $\overline{RD}$ să fie la un nivel logic care depinde numai de stările logice ale semnalelor de citit și de scris $\overline{MRD}$ și $\overline{MWR}$.

Figura 11 arată cum trebuie modificat circuitul din figura 8 pentru a răspunde precizărilor menționate. De obicei, acest circuit face parte integrantă din ansamblul-memorie și principiul său poate fi extins la un număr oarecare de intrări „Chip Select".

În concluzie, dacă dispunem de două memorii cu capacitatea de 1 kbit fiecare și dacă amîndouă memoriile sînt prevăzute cu o intrare $\overline{CS_1}$ și o intrare $\overline{CS_2}$, putem realiza o memorie cu capacitatea de 2 kbiți fără a folosi vreo componentă adițională, așa cum se poate vedea în **figura 12.**

Memoriile RAM pot fi de două tipuri: RAM static și RAM dinamic.

Memoriile RAM de tip static sînt alcătuite din celule a căror structură este cea a registrului LATCH, arătat în figura 5. Aceste memorii conservă conținutul lor atît timp cît ele sînt alimentate.

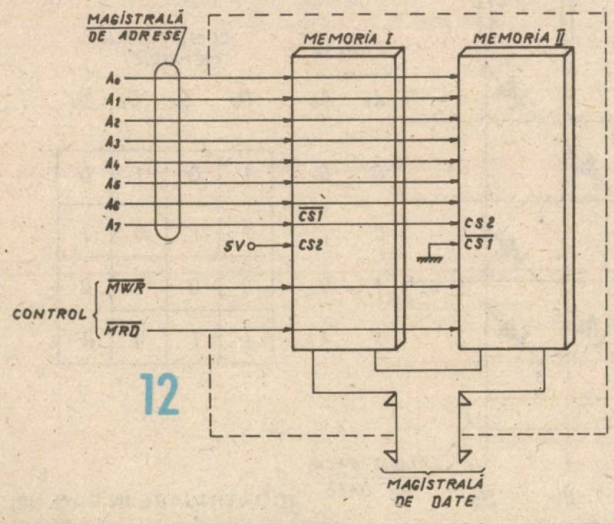
Memoriile RAM de tip dinamic folosesc un principiu de stocare a informațiilor cu totul diferit: este vorba de capacitatea existentă, între poartă și drenă, a unui tranzistor MOS, care are ca funcție memorizarea unui nivel logic. Structura ce intră în componența unei astfel de celule este deci extrem de simplă, ceea ce permite obținerea unei capacități de stocare relativ mare pentru un singur ansamblu-memorie. Tipic, pentru aceeași suprafață a plăcuței de siliciu folosite, o memorie RAM de tip dinamic poate avea o capacitate de 16 kbiți, în timp ce o memorie RAM de tip static prezintă numai o capacitate de 1 kbit. Pe de altă parte, memoriile RAM de tip dinamic conservă conținutul lor numai dacă pierderea naturală a sarcinii capacității unui tranzistor MOS este compensată prin reîncărcarea sa periodică. Se spune că este necesar să „reîmprospătăm" memoria; aceasta presupune existența unor circuite auxiliare ale căror complexitate și cost sînt justificate începînd cu o capacitate de ordinul a 32 kbiți. Sub această valoare este mult mai simplu și în același timp mai ieftin să asociem mai multe memorii RAM de tip static avînd capacități de stocare mai mici.

11

A_1	A_0	L_1	L_2	L_3	L_4
0	0	0	1	1	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0

12

BINAR	ZECIMAL	HEXAZECIMAL
0 0 0 0 0	0	0
0 0 0 0 1	1	1
0 0 0 1 0	2	2
0 0 0 1 1	3	3
0 0 1 0 0	4	4
0 0 1 0 1	5	5
0 0 1 1 0	6	6
0 0 1 1 1	7	7
0 1 0 0 0	8	8
0 1 0 0 1	9	9
0 1 0 1 0	10	A
0 1 0 1 1	11	B
0 1 1 0 0	12	C
0 1 1 0 1	13	D
0 1 1 1 0	14	E
0 1 1 1 1	15	F
1 0 0 0 0	16	10
1 0 0 0 1	17	11
1 0 0 1 0	18	12
1 0 0 1 1	19	13
1 0 1 0 0	20	14
— — — — —	----	----



MEMORIILE ROM

Denumirea ROM provine din limba engleză: „Reed Only Memory", ceea ce înseamnă „memorie numai la citire"; este vorba de memorii pe care constructorul le-a programat o dată pentru totdeauna, iar beneficiarul nu poate decît să le citească.

Structura unei memorii ROM este cea a unei matrice cu diode. **Figura 13** reprezintă o astfel de memorie cu capacitatea de 16 kbiți organizată în 4 linii și 4 coloane. Pentru o anumită adresă prezentă la intrarea decodorului linia L corespunzătoare este adusă la nivel logic 0, în timp ce celelalte sînt la nivel logic 1. În consecință, diodele ale căror catode sînt legate la linia selecționată sînt în stare de conducție, în timp ce celelalte rămîn blocate. Aceasta determină ca și coloanele legate la anodele diodelor aflate în stare de conducție să fie la nivel logic 1.

Ținînd seama de simplitatea structurii memoriilor ROM, capacitatea lor este relativ mare pentru un singur ansamblu (tipic: 8 la 16 kbiți).

MEMORIILE PROM

Denumirea PROM provine din limba engleză: „Programable ROM"; este vorba de o memorie ROM pe care beneficiarul o poate programa.

Structura unei memorii PROM este cea a unei memorii ROM, cu diferența că în serie cu diodele existente sînt dispuse fuzibile intacte la livrarea produsului. Memoria conține deci, în stadiul inițial, niveluri logice 0.

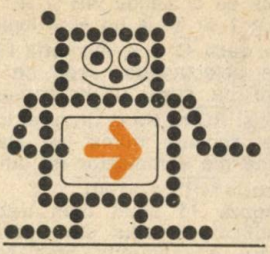
Dacă printr-un procedeu oarecare se pot distruge în mod selectiv aceste fuzibile, deci putem să le transformăm în circuite deschise, se va scrie nivelul logic 1 în memorie. Este evident că nu este posibil să modificăm erorile de programare.

MEMORIILE EPROM

Această denumire provine tot din limba engleză: „Erasable PROM", ceea ce înseamnă o memorie PROM la care utilizatorul poate să și „șteargă". Fiecare celulă este alcătuită dintr-un tranzistor MOS a cărui poartă este complet izolată **(fig. 14** și **15).** Principiul de memorare constă în folosirea stării de conducție sau de blocare a acestui tranzistor.

Trebuie de la început să se țină seama de următoarele precizări:

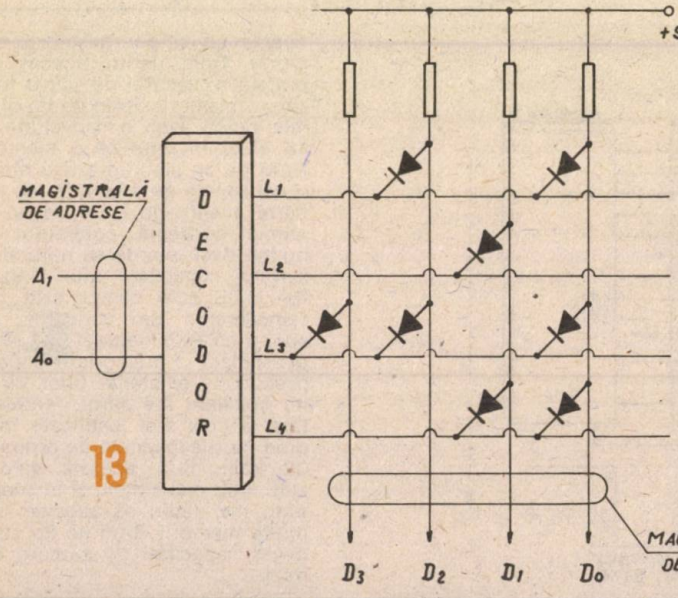
— dacă poarta G este lipsită de sarcini **(fig. 14),** totul se petrece ca și cum tranzistorul nu ar exista, deoarece circuitul cuprins între drenă și sursă este echivalent cu două diode înseriate avînd catodele legate împreună (tranzistorul este blocat);



— dacă poarta G conține un exces de electroni **(fig. 15),** electronii, ca purtători majoritari în substratul N, sînt respinși în jos, în timp ce golurile, ca purtători minoritari, sînt atrase de poartă. Între cele două zone dopate cu impurități de tip P se găsește deci indus un canal P care determină ca tranzistorul să fie în stare de conducție.

Cînd memoria EPROM este livrată de către constructor, porțile tuturor celulelor sînt lipsite de sarcini. Pentru a programa (a scrie) memoria, trebuie să găsim un mijloc pentru a încărca poarta negativ; aceasta pare imposibil deoarece, prin definiție, poarta este izolată.

În realitate siliciul nu este un izolant perfect și electronii cu energii mari sînt capabili să-l traverseze. Astfel, dacă joncțiunea PN constituită de sursa S și substratul N este polarizată invers de către o sursă cu tensiunea V suficient de mare (30 V la 40 V), atunci zona P **(fig. 16)** difuzează în substratul N electroni, dintre care unii au energii suficiente pentru a atinge poarta. Cînd tensiunea inversă este suprimată, poarta rămîne izolată, bitul corespunzător fiind programat.



13

ADRESA		CONȚINUTUL MEMORIEI			
A_1	A_0	D_3	D_2	D_1	D_0
0	0	1	0	1	0
0	1	1	1	0	1
1	0	0	0	1	0
1	1	1	1	0	0

(CONTINUARE ÎN PAG. 82)

LIMBAJUL FORTH

6

GHEORGHE CHIȚA

Revista "Tehnium" prezintă de cîtva timp în numerele sale microcalculatorul L/B881. În acest articol se va prezenta limbajul FORTH, recent implementat pe acesta.

Varianta de FORTH pentru L/B881 este compatibilă cu versiunea FIG-FORTH. Tot FIG-FORTH există și pentru microcalculatoarele M118 și ZX81. Limbajul mai este implementat și pe HC-85, Spectrum, Apple-II și Commodore 64, dar versiunile lor s-ar putea să nu fie FIG-FORTH. De aceea, în articol sînt prezentate lucruri suficient de generale pentru a funcționa pe orice versiune.

1. Noțiuni fundamentale

Pentru că tot trebuie să începem cu ceva, vom scrie un program care să nu facă altceva decît să spună ceva amabil atunci cînd este executat. În FORTH aceasta este foarte simplu: introduceți

```
." Salut, prietene! "
```

urmat de <cr> și aveți grijă să nu uitați vreun spațiu; FORTH va răspunde în mod agreabil

```
Salut, prietene! Ok
```

Prima observație: FORTH așteaptă comenzi pe care le execută și raportează "Ok" dacă totul merge bine sau afișează un mesaj de eroare în caz contrar (în acest caz lucrurile nu mai sînt "Ok"); încercați de exemplu să omiteți spațiul dintre ." și Salut.

Rezultă de aici a doua observație: spațiile sînt importante, căci ele separă **cuvintele**. În FORTH, comenzile și instrucțiunile se numesc "cuvinte" (words); nu confundați cu noțiunea de "unitate de memorie cu lungimea de 16 biți (2 octeți)" care în FORTH poartă denumirea de **celulă** (cell). Cuvintele sînt oarecum similare cu procedurile din alte limbaje, și așa cum acolo putem defini noi proceduri, în FORTH putem defini noi cuvinte - cu o mare deosebire însă - în FORTH noile definiții nu pot fi deosebite de cele vechi.

Să exemplificăm: CR este un cuvînt care trimite la terminal un <cr> urmat de <lf> astfel încît cursorul este adus la început de rînd nou. Definim un nou cuvînt SALUT, care va tipări mesajul de mai sus:

```
: SALUT   ." Salut, prietene! " ;<cr>
```

FORTH va răspunde "Ok" dacă n-ați uitat vreun spațiu sau punct și virgula de la sfîrșit și mai ales tasta "return" (o precizare: <cr> nu înseamnă să apăsați "<" apoi "c", "r" și la sfîrșit ">", ci semnifică apăsarea tastei "return"). "Ok" ce ? vă veți întreba. "Ok", FORTH a creat un nou cuvînt SALUT, care afișează mesajul dorit. Introduceți

```
SALUT<cr>
```

și veți obține un rezultat cunoscut

```
Salut, prietene! Ok
```

Acum vine partea interesantă: nu ne place ca răspunsul și "Ok"-ul să apară imediat după mesaj, ci le-am dori pe un rînd nou. Există două soluții:

- introducem

```
CR SALUT CR<cr>
```

și obținem

```
Salut, prietene!
Ok
```

Aceasta este soluția pe care n-o vom folosi, dar care dovedește că între o definiție nouă (SALUT) și una veche (CR) nu există nici-o deosebire. FORTH le găsește și le execută pe amîndouă. Asta probabil lămurește soluția a doua

- introducem

```
: SALUT   CR SALUT CR ;<cr>
```

și FORTH spune

```
SALUT Isn't unique Ok
```

adică mai există deja un cuvînt cu numele SALUT, dar lucrurile sînt "Ok"; în continuare nu va fi folosită decît cea mai recentă definiție, astfel

```
SALUT<cr>
Salut, prietene!
Ok
```

A treia observație: se poate adăuga o nouă definiție cu ":", și ea se termină cu ";". După ":" (în engleză "colon", de aceea definițiile de acest tip se numesc "colon-definitions") urmează numele noii definiții, apoi cuvintele care descriu definiția și la sfîrșit ";" (semicolon).

În acest moment vă puneți deja o serie de întrebări, de exemplu: ce faceți dacă introduceți ceva greșit - cum corectați ? Foarte simplu: ultimul caracter poate fi șters cu <delete> sau <backspace>; toată linia poate fi anulată cu <ctrl/X>. Ați înțeles probabil că toate corecturile trebuie făcute înainte de a apăsa <cr>, căci după aceea FORTH va încerca să execute ceea ce ați introdus.

Precizări înainte de a trece la întrebarea următoare: literele mari și mici sînt absolut echivalente (deși în acest text se folosesc numai litere mari pentru a nu creea confuzii, se recomandă folosirea aproape întotdeuna a literelor mici, este mai estetic; pe FORTH oricum nu-l

6

interesează, el trece totul la litere mari). Deasemenea, numele unui cuvint poate conţine şi cifre şi semne speciale (in orice ordine doriţi) şi chiar caractere de control (!), cu excepţia lui <delete>, <backspace>, <ctrl/X> şi <cr>. De exemplu, există un cuvint care este <ctrl/@>, puteţi să-l executaţi dacă nu credeţi.

Intrebare: "Ce se intimplă dacă se introduce SALIT in loc de SALUT ?". Puteţi incerca astfel

SALIT<cr> SALIT ?

Vedeţi că acum nu mai este "Ok". Acum apare o altă intrebare - de unde ştie că SALIT nu e corect ? La urma urmei de unde ştie că SALUT este corect ?

Ajungem astfel la un alt concept important - **dicţionarul** (dictionary). Intrucit comenzile FORTH sint cuvinte, este evident că locul lor este intr-un dicţionar. Iniţial, acesta conţine doar cuvintele de bază care alcătuiesc **nucleul** (kernel) sistemului. Pe măsură ce adăugăm noi cuvinte (prin ":" sau prin alte mijloace) ele sint trecute - impreună cu definiţiile lor - la sfirşitul dicţionarului. Cind FORTH incepe să caute cuvintele care i le daţi spre execuţie, el incepe cu ultimul cuvint definit şi merge spre inceputul dicţionarului. Această tehnică economiseşte timpul de căutare căci se presupune că intr-o nouă definiţie vă bazaţi mai mult pe propriile voastre definiţii decit pe cele din nucleu (care sint la inceputul dicţionarului, evident). Aceasta este ideea fundamentală a FORTH-ului (fraza următoare ar trebui să o subliniaţi cu creion roşu): o aplicaţie scrisă in FORTH porneşte cu un număr de cuvinte de bază, pe care se clădesc cuvinte mai puternice, pe care se construiesc cuvinte şi mai puternice şi tot aşa pină se ajunge la rezolvarea problemei pusă de aplicaţia respectivă.

Să ilustrăm aceasta printr-un mic exemplu (sugerat de Leo Brodie in cartea sa "Starting FORTH"): dorim să tipărim o literă mare F (de la FORTH) din asterisk-uri care să arate cam aşa:

```
*****
*
*****
*
*
```

Intii ne punem problema tipăririi unui asterisk. Aceasta se poate face astfel

42 EMIT<cr> *Ok

Cum funcţionează veţi inţelege mai tirziu, acum constatăm doar efectul: după ce apăsăm <cr> apare imediat steluţa şi totul este "Ok". Intrucit ne este de folos, introducem comenzile intr-o definiţie şi obţinem un nou cuvint

: STAR 42 EMIT ;<cr> Ok

Verificăm incă o dată

STAR<cr> *Ok

Acum trebuie să tipărim mai multe steluţe la rind, deci

: STARS 0 DO STAR LOOP ;<cr> Ok

Acestei definiţii ii transmitem un număr (vom vedea cum) şi ea va tipări cite steluţe vrem. Se vede că folosim STAR pe care tocmai l-am definit şi incă două cuvinte care probabil că aţi inţeles că sint echivalentul lui FOR-NEXT din BASIC. Priviţi-l pe STARS in acţiune

5 STARS<cr> *****Ok

Puteţi incerca şi ceva de genul 10000 STARS dar pentru asta trebuie să aveţi ceva răbdare.

Acum să vedem cum aducem litera in mijlocul ecranului. Pentru aceasta trebuie să tipărim un număr de spaţii astfel

15 SPACES<cr> Ok

Deci vom defini

: MARGIN CR 20 SPACES ;<cr> Ok

Facem citeva experienţe pentru a căpăta incredere

MARGIN STAR <cr>

şi obţinem

* Ok

sau

MARGIN 5 STARS <cr>

rezultind

***** Ok

Putem crea deci incă două definiţii

: BLIP MARGIN STAR ;<cr> Ok
: BAR MARGIN 5 STARS ;<cr> Ok

Si acum finalul

: F BAR BLIP BAR BLIP BLIP CR ;<cr> Ok
F<cr>

```
*****
*
*****
*
*
```

Ok

Pentru claritate, iată cum arată listingul aplicaţiei noastre (cum obţinem listingul veţi vedea mai incolo)

```
0 ( Litera mare F )
1 : STAR  42 EMIT ;
2 : STARS  0 DO STAR LOOP ;
3 : MARGIN  CR 20 SPACES ;
4 : BLIP  MARGIN STAR ;
5 : BAR  MARGIN 5 STARS ;
6 : F  BAR BLIP BAR BLIP BLIP ;
```

Iată deci că am şi scris primul nostru program in FORTH, prilej cu care am invăţat cite ceva despre dicţionar. Să studiem acum problema interpreterului FORTH, şi in conjuncţie cu ea, compilatorul.

Ce se intimplă cind introduceţi ceva de genul

STAR 30 SPACES<cr>

Iată cam cum se desfăşoară lucrurile: după ce apăsaţi <cr>, intră în acţiune un cuvînt care se cheamă INTERPRET, cunoscut şi sub numele de interpreter de text sau interpreterul exterior. El baleiază textul introdus, căutînd şiruri de caractere separate de spaţii. Cînd găseşte un astfel de şir, îl caută în dicţionar. Dacă-l găseşte, transmite definiţia acestuia unui cuvînt numit EXECUTE, care execută definiţia (în acest caz tipăreşte un asterisk). Dacă nu-l găseşte în dicţionar, îl pasează unui cuvînt ce se ocupă de numere (numit NUMBER). NUMBER ştie cum trebuie să arate un număr. Dacă ceea ce i s-a transmis este un număr, el este depus într-o locaţie specială (vom vedea îndată care). Se ajunge astfel la SPACES, care este executat (pentru că este în dicţionar) rezultînd numărul corespunzător de spaţii. În final, INTERPRET dă de <cr> care-i spune că totul s-a terminat cu bine, deci va apare "Ok".

Vă mai aduceţi aminte de SALIT ? Ce se întîmplă în acest caz ? Ei bine, SALIT nefiind în dicţionar, INTERPRET îl transmite lui NUMBER, dar acesta vede că nu este vorba de un număr; drept urmare el raportează eroare (arătîndu-ne vinovatul urmat de un semn de întrebare), deci lucrurile nu sînt "Ok".

Pînă acum am avut de-a face numai cu INTERPRET, FORTH fiind tot timpul în **starea de execuţie** (executing). FORTH poate fi şi în **starea de compilare** (compiling), de exemplu cînd introducem

```
: STAR   42 EMIT ;<cr>
```

După tastarea lui <cr>, INTERPRET porneşte la lucru (sîntem în starea de execuţie). El găseşte ":", care este un cuvînt în dicţionar, şi îl trimite lui EXECUTE. Prin execuţie ":" porneşte procesul de compilare (FORTH trece astfel în compilare), care înscrie noua definiţie în dicţionar, într-o formă corespunzătoare. Cînd este executat de EXECUTE, ";" opreşte compilatorul şi dă controlul lui INTERPET (deci din nou în execuţie), care, dînd de <cr>, ne transmite "Ok".

Să recapitulăm un pic înainte de a trece mai departe: cînd introduceţi cuvinte de la terminal, ele sînt interpretate şi apoi executate (FORTH este în "execuţie"). Cînd definiţi un cuvînt nou (cu ":"), FORTH trece în "compilare"; definiţia este "compilată" în dicţionar - nu se execută nimic din ea acum - apoi FORTH revine la execuţie.

Să vedem acum ce este cu locaţia misterioasă unde NUMBER trimite numerele. Se presupune că sînteţi familiarizaţi cu o stivă; pentru aceia care nu ştiu ce este, iată aici o scurtă descriere: o "stivă" (stack) este o succesiune de locaţii de memorie la care se face acces după regula ultimul-venit-primul-servit (Last-In-First-Out); puteţi face analogie cu o stivă de farfurii proaspăt spălate: cînd le puneţi la loc în dulap, începeţi cu ultima spălată şi nu puteţi ajunge la prima decît după ce le-aţi îndepărtat pe toate celelalte. Spunem că la o stivă nu avem acces, la un momemt dat, decît la vîrful ei (top of stack). Există o locaţie specială care indică întotdeauna vîrful stivei şi care se numeşte stack-pointer (indicatorul stivei). Să vedem ce legătură au toate acestea cu FORTH-ul. Introduceţi

```
3 4 + .<cr>
```

şi obţineţi (era de aşteptat, nu ?)

```
7 Ok
```

pe aceeaşi linie. Iată ce se întîmplă: cînd introduceţi un număr de la terminal, text interpreter-ul îl transmite lui NUMBER, care îl pune într-o anumită locaţie. Acea locaţie este vîrful stivei, deci cînd dă de "3", NUMBER pune în vîrful stivei valoarea trei. Apoi se ajunge la "4", care este trimis şi el în vîrful stivei "împingîndu-l" pe trei mai jos (în realitate nimic nu se mişcă; "4"-ul este pus lîngă "3", iar stack-pointer-ul arată acum spre "4"). Următorul cuvînt din şirul de la intrare se găseşte în dicţionar. "+" a fost anterior definit astfel: "ia cele două numere din vîrful stivei, adună-le, şi pune rezultatul la loc în stivă". Următorul cuvînt este deasemenea în dicţionar. El a fost definit să ia numărul din vîrful stivei şi să-l afişeze la terminal.

Bine, veţi zice, dar ce este cu "+"-ul acesta ciudat pus după numere şi nu între ele cum ar fi fost normal ? FORTH foloseşte notaţia "postfix" (numită aşa pentru că operatorul se pune **după** operanzi) în loc de uzuala notaţie "infix" (în care operatorul apare **între** operanzi). Această tehnică permite ca orice cuvînt care are nevoie de numere să le ia din stivă, iar dacă furnizează alte numere ca rezultat, să le pună tot în stivă. De exemplu

- + ia două numere din stivă, le adună, iar rezultatul îl pune tot în stivă;
- . ia un număr din stivă şi îl tipăreşte;
- SPACES ia un număr din stivă şi tipăreşte un număr egal de spaţii;
- EMIT ia un număr ce reprezintă un caracter şi tipăreşte acel caracter;
- STARS, pe care l-am definit mai sus, ia un număr din stivă şi tipăreşte un număr egal de asterisk-uri.

Cînd toţi operatorii au fost definiţi să lucreze pe numere ce există deja în stivă, interacţiunea dintre ei rămîne simplă chiar atunci cînd programul devine complex. Acest lucru vă va fi tot mai clar pe măsură ce ne vom continua studiul FORTH-ului. Înainte de a trece mai departe însă, vom introduce o notaţie simplă a efectelor cuvintelor asupra stivei (este clar că avem nevoie de aşa ceva, altfel nu vom şti de ce parametrii are nevoie fiecare cuvînt şi ce rezultate produce). Iată cum vom face

```
( înainte -- după )
```

Cele două liniuţe separă lucrurile ce existau în stivă înaintea execuţiei cuvîntului de lucrurile rămase după execuţie. Poziţia în stivă este indicată de ordinea în care scriem: obiectul **cel mai din dreapta** este, în ambele cazuri, cel aflat în vîrful stivei. Exemplu pentru "."

```
( n -- )
```

"." (care se pronunţă "dot") aşteaptă un număr ("n" este de la "număr") şi nu returnează nimic.
Pentru "+" lucrurile stau astfel

```
( n1 n2 -- suma )
```

Aici n2 este în vîrf şi n1 sub el; se returnează o valoare care este suma celor două numere.

Să mai discutăm un pic despre stivă. Dacă introducem

```
1 2 3 4<cr> Ok
```

și acum încercăm să tipărim numerele (cu "dot"), în ce ordine credeți că vor apare ? Nu știți ? Gîndiți-vă un pic, o stivă este ceva de genul ultimul-intrat-primul-ieșit. Care este ultimul intrat ? Patru. Sub el este trei, apoi doi și în sfîrșit unu. Introducem

```
. . . .<cr> 4 3 2 1 Ok
```

și obținem ceea ce ne așteptam. Puteți să faceți și așa

```
1 2<cr> Ok
3 4<cr> Ok
.<cr> 4 Ok
. . .<cr> 3 2 1 Ok
```

Vedeți deci că stiva este un loc în care puteți avea încredere, lucrurile rămîn acolo dacă nu le scoateți - cu o excepție: cînd FORTH nu semnalizează "Ok" el golește automat stiva.

Tot legat de stivă, există două situații speciale. Cînd încercați să scoateți din stivă mai multe elemente decît sînt, se produce "stack underflow" ("subdepășire stivă") și FORTH spune "Stack empty" ("stiva goală"); aceasta nu-l deranjează. Dacă însă introduceți mai multe elemente în stivă decît este loc, FORTH vă va spune "Stack full" ("stiva plină"); aceasta poate să-l deranjeze și este recomandat să reîncărcați FORTH-ul. Capacitatea stivei este însă foarte mare și de aceea acest ultim mesaj nu-l veți vedea decît dacă faceți ceva foarte rău.

Iată-ne ajunși la sfîrșitul secțiunii introductive. Mai departe vom intra în amănunte despre diversele aspecte ale limbajului, folosind intensiv exemple. Pentru a economisi spațiu, nu se va mai arăta separat ceea ce se introduce de la terminal și ce răspunde FORTH, ci se vor scrie în continuare, separate de <cr> (simbolul tastei "return").

2. Editorul

După cum ați văzut, textul introdus de la claviatură se pierde după execuție; dacă vreți să corectați ceva la el trebuie introdus din nou.

FORTH organizează memoria externă în blocuri de cîte 1024 octeți (exact 1K), numite chiar **blocuri** (blocks) sau **ecrane** (screens). Fiecare bloc este privit ca avînd 16 rînduri (numerotate de la 0 la 15) cu 64 octeți fiecare.

Textul se introduce în aceste blocuri folosind **editorul** (editor). Metodele de editare sînt specifice sistemului pe care este implementat FORTH; editorul lui 881/Forth a fost scris astfel încît comenzile să fie (aproape) identice cu cele ale editorului din 881/Sys.

Odată introdus, textul poate fi interpretat dintr-un bloc ca și cum ar fi introdus de la claviatură. Pentru aceasta se folosește comanda LOAD. LOAD (care înseamnă "încarcă") așteaptă în stivă un număr care să-i spună ce bloc să interpreteze. Mai departe totul decurge normal, cu o mică excepție: în caz de eroare puteți introduce WHERE (unde) și editorul va fi pornit automat cu cursorul poziționat pe cuvîntul care se presupune că a comis eroarea.

Conținutul unui bloc poate fi listat cu comanda LIST (listează), acest cuvînt așteptînd în stivă numărul blocului de listat. Se poate folosi L pentru a lista blocul curent (deci nu mai este nevoie să specificați numărul lui), P (previous = anterior) pentru blocul anterior acestuia și N (next = următor) pentru blocul următor blocului curent.

Nu se vor explica comenzile editorului, acestea fiind descrise în manualul lui 881/Forth, mai ales că utilitatea lor pentru cei care au alte sisteme de FORTH este nulă (aceștia trebuie să citească manualele propriilor lor sisteme).

Vom insista puțin asupra felului cum trebuie organizat textul sursă. Acesta este un lucru important, deoarece text sursă FORTH scris neordonat arată cam ca o telegramă cifrată, numai autorul poate înțelege ceva. De aceea s-au elaborat cîteva reguli de stil (rezultate dintr-o practică îndelungată) pentru comentarea unui program.

În acest context al comentariilor sînt utile următoarele cuvinte:

- ;S ("skip-screen" = "sări restul ecranului") oprește interpretarea blocului. După el puteți pune orice text, el nu va fi luat în considerare, deci puteți introduce comentarii.
- \ ("skip-line" = "sări restul liniei") ignoră restul liniei. Se folosește în cadrul unei definiții pentru a comenta părțile mai obscure.
- (("left-paren" = "paranteză deschisă") ignoră textul pînă la apariția unei ")" ("right-paren" = "paranteză închisă"). Utilă pentru comentariile efectelor asupra stivei, astfel:

```
: STARS   ( n -- )  ... ; \ tipărește n steluțe
```

Și acum, cîteva reguli de stil :

- Rezervați linia 0 (prima linie) a oricărui bloc pentru o descriere a blocului; începeți cu "\", astfel ca FORTH să ignore restul liniei, iar în ultimele poziții ale liniei notați data și autorul programului.
- Scrieți o singură definiție pe linie, chiar dacă este loc pentru mai multe. Definiția se poate continua pe rîndurile următoare, care vor fi totdeauna identate cu trei spații.
- După numele cuvîntului nou definit lăsați trei spații apoi scrieți efectul asupra stivei, apoi încă trei spații și descrieți pe scurt ce face definiția (dacă nu este suficient de clar din textul ei).
- Definițiile trebuie să fie scurte, de maximum trei linii. În cadrul lor separați frazele prin două spații.

Ar mai fi multe de spus aici, dar expunerea ar fi lipsită de conținut atît timp cît nu ați învățat FORTH. În orice caz urmăriți respectarea regulilor de mai sus în exemplele ce vor urma.

3. Numere

Nu vă speriați căci nu vom începe un curs de matematică, deși o parte a acestei secțiuni arată un pic ca un curs de algebră. Numerele sînt însă importante pentru că ele modelează realitatea înconjurătoare într-o formă accesibilă calculatorului. Un număr poate reprezenta o adresă de memorie, un caracter grafic, o notă muzicală, o stare sau orice altceva doriți. De aceea studiul lor este important, chiar dacă vreți să rezolvați cu (micro-) calculatorul dumneavoastră probleme nenumerice. Deci, nu ezitați, citiți mai departe!

Numerele întregi (integers) pot fi reprezentate în FORTH pe orice lungime, dar cel mai des se folosesc 16 biți (numere de lungime simplă = single length numbers) și 32 de biți (numere de lungime dublă = double length numbers). Numerele

de lungime simplă pot fi cu semn (signed) sau fără semn (unsigned). Primele sint notate cu "n" in comentariile stivelor, iar celelalte cu "u" (de la unsigned). Similar, numerele de lungime dublă pot fi cu semn (notate cu "d") sau fără semn (notate cu "ud").

Intregii cu semn pe 16 biţi pot lua orice valori intre -32768 şi 32767. Iată o listă de operatori cu care putem transforma FORTH-ul intr-un calculator cu notaţie postfix:

```
+   ( n1 n2 -- suma )          citiţi "plus"
    suma = n1 + n2
-   ( n1 n2 -- dif )                  "minus"
    dif = n1 - n2
*   ( n1 n2 -- prod )                 "star"
    prod = n1 * n2
/   ( n1 n2 -- cit )                  "slash"
    cit = n1 / n2
```

Observaţi că trecerea de la infix la postfix este destul de simplă: pur şi simplu mutaţi operatorul dintre operanzi după ei. Dacă doriţi să aflaţi cit este 5+7 introduceţi

```
5 7 + .<cr> 12 Ok
```

Alte exemple

```
6 -8 * .<cr> -48 Ok
12 4 / .<cr> 3 Ok
9 2 - .<cr> 7 Ok
```

Ce faceţi cu ceva de genul 4*(5+6) ? Simplu, parantezele vă spun să faceţi intii adunarea şi apoi inmulţirea

```
5 6 + 4 * .<cr> 44 Ok
```

Notaţia postfix are o proprietate remarcabilă şi anume nu are nevoie de paranteze; introduceţi operanzii in ordinea in care se află in expresie, ignorind parantezele, şi apoi aplicaţi operatorii intr-o ordine convenabilă. Astfel, exemplul de mai sus poate fi scris

```
4 5 6 + * .<cr> 44 Ok
```

Dacă n-aţi inţeles cum funcţionează, iată aici soluţia:

Intrare	Stiva (cu virful spre dreapta)
4	4
5	4 5
6	4 5 6
+	4 11
*	44
.	stiva este goală

Incă un exemplu inainte de a trece mai departe. Avem de calculat (9+5)*(7+4). Se rezolvă astfel (uitaţi-vă la regula de mai sus)

```
9 5 + 7 4 + * .<cr> 154 Ok
```

Este recomandabil să desenaţi o diagramă a stivei şi intrării ca cea dinainte pentru a inţelege mai bine.

Să discutăm puţin despre impărţire. Cit credeţi că este 22/3 ?

```
22 3 / .<cr> 7 OK
```

Trebuia să vă aşteptaţi la asta, deoarece slash, ca şi plus, minus şi star lucrează cu intregi. De aici nu rezultă insă că nu putem face socoteli "cu virgulă", trebuie doar să folosim metode mai avansate.

Putem afla şi restul impărţirii a doi operanzi folosind

```
/MOD   ( n1 n2 -- n-rest n-cit )     "slash-mod"
MOD    ( n1 n2 -- n-rest )           "mod"
```

```
37 4 /MOD . .<cr> 9 1 Ok
```

Deoarece citul era in virf, el apare primul la tipărire.

```
37 4 MOD .<cr> 1 Ok
```

Ca o aplicaţie, să presupunem că avem de convertit un număr de luni in ani. Putem face astfel

```
: LUNI   ( n -- )   12 /MOD
   . ." ani şi " . ." luni " ;
```

Testăm astfel

```
30 LUNI<cr> 2 ani şi 6 luni Ok
```

Să trecem mai departe considerind următoarea situaţie: trebuie să scriem o definiţie care să calculeze (a-b)/c dindu-se la intrare numerele c a b in această ordine. Vom constata că nu putem rezolva această problemă decit dacă putem schimba intr-un fel ordinea numerelor in stivă. Iată ciţiva operatori care fac acest lucru

```
SWAP   ( n1 n2 -- n2 n1 )                "swap"
       Inversează primele două elemente din stivă
DUP    ( n -- n n )                      "dup"
       Duplică virful stivei
OVER   ( n1 n2 -- n1 n2 n1 )             "over"
       Copiază al doilea element
       şi il pune in virf
ROT    ( n1 n2 n3 -- n2 n3 n1 )          "rot"
       Roteşte al treilea element in virf
DROP   ( n -- )                          "drop"
       "Aruncă" elementul din virf
```

Folosind SWAP problema se rezolvă astfel

```
: RESULT   ( c a b -- )   - SWAP / . ;
```

Pentru a=10 b=4 şi c=3 obţinem

```
3 10 4 RESULT<cr> 2 Ok
```

DUP este necesar cind trebuie calculat a^2

```
5 DUP * .<cr> 25 Ok
```

Dacă avem nevoie de a^2+a*b, cu a b in stivă, atunci scriem a^2+a*b = a*(a+b) şi introducem (pentru a=5 şi b=7)

```
5 7 OVER + * .<cr> 60 Ok
```

Pentru a calcula a*b-b*c avind la intrare c b a, scriem intii expresia ca b*(a-c) şi o calculăm cu

```
ROT - *
```

Următorul tabel prezintă nişte operatori foarte simpli, dar a căror existenţă este bine justificată. Sint trei motive pentru a scrie "1+" in loc

de "1" şi "+". Unu: se face economie de spaţiu in dicţionar pentru că sînt folosiţi foarte des de programator şi chiar de către sistemul FORTH. Doi: compilarea este puţin mai rapidă. Trei: se execută mult mai repede; de exemplu, "2* 2* 2*" este mai rapid decit "8 *" şi asta poate fi uneori mai important decit risipa de memorie.

```
1+   ( n -- n+1 )                    "unu-plus"
     Adună unu
1-   ( n -- n-1 )                    "unu-minus"
     Scade unu
2+   ( n -- n+2 )                    "doi-plus"
     Adună doi
2-   ( n -- n-2 )                    "doi-minus"
     Scade doi
2*   ( n -- n*2 )                    "doi-star"
     Inmulţeşte cu doi
     (deplasare aritmetică stinga)
2/   ( n -- n/2 )                    "doi-slash"
     Imparte la doi
     (deplasare aritmetică dreapta)
```

In incheiere, o tabelă cu cîţiva operatori interesanţi

```
MINUS   ( n -- -n )                  "minus"
        Schimbă semnul
ABS     ( n -- |n| )                 "absolut"
        Valoarea absolută
MIN     ( n1 n2 -- n-min )           "min"
        Returnează minimul
MAX     ( n1 n2 -- n-max )           "max"
        Returnează maximul
```

De exemplu, dacă vrem să aflăm maximum dintre trei numere aflate in stivă, procedăm astfel

```
4 -5 9 MAX MAX .<cr> 9 Ok
```

Punem punct acestui prim capitol despre numere. Vom mai vorbi insă despre ele şi vom studia alte cuvinte intr-o secţiune viitoare.

4. Structuri de control

In această secţiune vom studia modul in care putem lua decizii şi modul in care putem executa o sarcină in mod repetitiv. Acestea sint două dintre lucrurile care fac calculatoarele să fie atit de folositoare.

Structurile de control nu pot fi testate direct, in modul de execuţie, ci trebuie introduse intr-o definiţie.

Deciziile se codifică astfel in FORTH

```
condiţie IF acţiunea1 ELSE acţiune2 THEN
```

Prin "condiţie" se inţelege o succesiune de cuvinte care produce in stivă o valoare logică (sau "flag" cum se mai numeşte). Dacă aceasta are valoarea "adevărat" atunci se execută secvenţa "acţiunea1", altfel se execută secvenţa "acţiune2". In ambele cazuri se continuă cu instrucţiunile de după THEN. Se poate folosi şi forma prescurtată

```
condiţie IF acţiune THEN
```

In această formă nu se execută nimic dacă flag-ul are valoarea fals.

Să presupunem că in stivă obţinem un număr ce reprezintă viteza in km/h a unui autovehicul in localitate. Putem vedea dacă este o viteză legală astfel

```
: ?LEGAL   60 > IF ." Viteză nelegală " THEN ;
55 ?LEGAL <cr> Ok
72 ?LEGAL <cr> Viteză nelegală Ok
```

">" compară primele două numere din stivă şi produce o valoare logică. Vedeţi că şi aici se foloseşte notaţia postfix: "55 > 60" se scrie "55 60 >", trecind operatorul după operanzi.

O valoare logică se reprezintă tot printr-un număr, "zero" insemnind "fals" şi "unu" insemnind "adevărat". Putem verifica astfel

```
55 60 > .<cr> 0 Ok
72 60 > .<cr> 1 Ok
```

IF, ca şi alte cuvinte care aşteaptă un flag, consideră insă tot ceea ce este diferit de zero drept adevărat, ceea ce este foarte convenabil. De exemplu

```
: TEST   IF ." adevărat " ELSE  " fals " THEN ;
9 TEST<cr> adevărat Ok
-5 TEST<cr> adevărat Ok
0 TEST<cr> fals Ok
```

Iată o primă listă de operatori care produc valori logice. In acest tabel, "f" inseamnă "flag".

```
=    ( n1 n2 -- f )  .               "egal"
     f = adevărat dacă n1 = n2
<    ( n1 n2 -- f )                  "mai-mic"
     f = adevărat dacă n1 < n2
>    ( n1 n2 -- f )                  "mai-mare"
     f = adevărat dacă n1 > n2
0=    ( n -- f )                     "zero-egal"
     f = adevărat dacă n = 0
0<    ( n -- f )                     "zero-mai-mic"
     f = adevărat dacă n < 0
```

Observaţi că nu există "diferit" şi vă intrebaţi ce faceţi dacă aveţi nevoie de el. Proprietatea lui IF expusă mai sus face ca acest operator să nu fie necesar; puteţi folosi "-" in loc.

```
: SINT   - IF ." diferite "
   ELSE ." egale " THEN ;<cr> Ok
34 76 SINT<cr> diferite Ok
5 5 SINT<cr> egale Ok
```

Acelaşi lucru este valabil şi pentru "zero-diferit"; aici nu este nevoie de nici-un operator, numărul insuşi putind fi folosit pe post de flag.

```
: ESTE   IF ." nenul " ELSE ." nul " THEN ;
5 ESTE<cr> nenul Ok
-8 ESTE<cr> nenul Ok
0 ESTE<cr> nul Ok
```

Iată acum un operator ciudat, care are un IF construit inăuntru. El duplică un număr numai dacă acesta este diferit de zero. Utilitatea sa nu este evidentă, dar cu timpul il veţi aprecia.

```
?DUP   ( n -- n n )                  "question-dupe"
         sau          Duplică doar dacă n
       ( 0 -- 0 )     este diferit de zero
```

Valorile logice pot fi combinate intre ele folosind operatorii

```
AND   ( f1 f2 -- f )              "and"
      f = adevărat dacă
      f1 şi f2 sint adevăraţi
OR    ( f1 f2 -- f )              "or"
      f = adevărat dacă
      f1 sau f2 este adevărat
XOR   ( f1 f2 -- f )              "exclusive-or"
      f = adevărat dacă
      f1 şi f2 sint diferiţi
```

In acest tabel f1, f2 şi f desemnează valori logice. Totuşi, aceşti operatori pot fi folosiţi şi cu numere obişnuite, pentru că operaţiile logice se fac bit cu bit (conform regulilor logicii boolene).

Să trecem acum la structuri repetitive, care permit implementarea ciclurilor (sau buclelor - in engleză "loops"). Acestea sint de două feluri: cu un număr cunoscut de repetări şi, evident, cu un număr necunoscut de repetări. In acest ultim caz ieşirea din ciclu se face in urma indeplinirii unei condiţii logice.

Construcţia

```
m n DO acţiuni LOOP
```

va executa "acţiuni" de m - n ori dacă acest număr este pozitiv, sau numai odată in caz contrar. Să incercăm un exemplu

```
: TEST  5 0 DO I . LOOP ;
```

DO-LOOP, ca şi IF-THEN, trebuie folosite numai in interiorul unei definiţii pentru că implică salturi de la un cuvint la altul. Cuvintul "I" aduce in stivă indexul curent al buclei, şi astfel obţinem

```
TEST<cr> 0 1 2 3 4 Ok
```

Vedeţi cum instrucţiunile dintre DO şi LOOP s-au executat de 5 - 0 = 5 ori. Dacă incercăm

```
: TEST1  0 5 DO I . LOOP ;<cr> Ok
TEST1<cr> 5 Ok
```

observăm că prin buclă se trece o singură dată (0 - 5 = -5). Putem incerca şi

```
: TEST2  5 5 DO I . LOOP ;<cr> Ok
TEST2<cr> 5 Ok
```

In acest caz 5 - 5 = 0 şi bucla se execută tot o singură dată.

"m" de mai sus se numeşte limita, iar "n" indexul buclei. Observaţi că DO-LOOP nu trece niciodată peste limită.

Pentru a inţelege cum funcţionează DO-LOOP trebuie să discutăm despre o altă stivă. Pină acum nu am vorbit decit despre "stivă" ca şi cum nu ar mai fi existat alta. Numele său complet este insă "parameter stack" ("stiva de parametrii"), şi i se spune aşa pentru că este folosită la transmiterea parametrilor intre diverse cuvinte. Cind nu există prilej de dubiu o numim simplu "stivă".

Cea de-a doua stivă se numeşte "return stack" ("stiva de intoarcere"). Ea este folosită la transmiterea controlului de la un cuvint la altul. Cu respectarea unor mici restricţii, ea poate fi privită ca un loc in care păstrăm valori temporare. Nu o putem folosi insă la transmiterea de parametrii intre cuvinte. In cadrul unei definiţii efectul asupra lui return stack trebuie să fie nul, tot ce am pus trebuie să fie scos inainte de ";" (deoarece ";" aşteaptă in return stack un pointer spre cuvintul care a invocat definiţia curentă).

Iată o listă de operatori folosiţi in legătură cu return stack. Nu uitaţi, comentariile din paranteză se referă la stiva obişnuită (parameter stack).

```
>R   ( n -- )                     "to-R"
     Ia un număr din parameter stack
     şi il pune in return stack.
R>   ( -- n )                     "R-from"
     Ia un număr din return stack
     şi il pune in parameter stack.
I    ( -- n )                     "I"
     Copiază virful lui return
     stack fără să-l modifice.
I'   ( -- n )                     "I-prim"
     Copiază al doilea element
     din return stack fără să-l modifice.
J    ( -- n )                     "J"
     Copiază al treilea element
     din return stack fără să-l modifice.
```

Un exemplu de aplicare pentru >R şi R> este cuvintul WITHIN (intre) care primeşte in stivă un număr şi două limite astfel

```
( n low high )
```

şi care verifică dacă low <= n < high. Aceasta se codifică astfel

```
: WITHIN   >R 1- OVER < SWAP R> < AND ;
```

Iată cum funcţionează

Intrare	Parameter stack (cu virful spre dreapta)	Return stack (idem)
	n low high	
R	n low	high
1-	n low-1	high
OVER	n low-1 n	high
<	n (low-1 < n)	high
SWAP	(low-1 < n) n	high
R>	(low-1 < n) n high	
<	(low-i < n) (n < high)	
AND	flag	

Expresiile din paranteze semnifică fiecare cite un flag. AND de la sfirşit verifică indeplinirea ambelor condiţii simultan. Observaţi cum nu se lasă şi nu se scoate nimic din return stack in afara definiţiei (efectul pe ansamblu este nul). >R şi R> pot fi testaţi chiar de la consolă cu aceeaşi precizare: efectul asupra lui return stack să fie nul inainte de a apăsa <cr>.

Să revenim acum la DO şi să vedem cum funcţionează. DO aşteaptă in parameter stack două valori: in virf limita şi dedesubt valoarea iniţială a indexului. Aceste două valori sint scoase din parameter stack şi trecute in return stack, in virf fiind indexul şi dedesubt limita. Apoi controlul este trecut cuvintelor de după DO. Cind se ajunge la LOOP, acesta verifică dacă indexul este mai mic decit limita şi in caz afirmativ măreşte indexul cu 1 şi redă controlul primului cuvint de după DO. In cazul cind indexul a ajuns limita, el scoate ambele valori din return stack şi le aruncă, apoi dă controlul cuvintelor de după el. Observaţi cum şi DO-LOOP respectă regula efectului nul asupra lui return stack.

Acum înțelegeți cum merge și I. Ceea ce copiază el din virful lui return stack se intimplă să fie tocmai indexul curent, pe care noi il tipărim cu dot in exemplele anterioare.

De asemenea, remarcați că in cadrul unei bucle DO-LOOP trebuie din nou respectată regula efectului nul (trebuie balansate R> cu >R), altfel ii distrugeți indexul și/sau limita și n-o să știe ce să mai facă.

I' il puteți folosi pentru a examina limita in cadrul lui DO-LOOP (deoarece acesta este al doilea element din virful lui return stack) - de exemplu cind aceasta este o valoare calculată prin program și nu transmisă explicit.

J este folositor cind aveți un DO inclus in alt DO, căci al treilea element din virful lui return stack este chiar indexul buclei exterioare.

Iată, dacă doriți să tipăriți o tablă a inmulțirii, puteți proceda astfel

```
: TABLA   CR
   11 1 DO  11 1 DO I J * . LOOP  CR LOOP ;
TABLA<cr>
1 2 3 4 5 6 7 8 9 10
2 4 6 8 10 12 14 16 18 20
3 6 9 12 15 18 21 24 27 30
.....................
10 20 30 40 50 60 70 80 90 100
Ok
```

In bucla interioară I merge de la 1 la 10, care inmulțit cu indexul buclei exterioare (furnizat de J) ne dă tabla inmulțirii pentru o singură cifră. Repetind totul de zece ori obținem rezultatul dorit.

Există un LOOP care ne permite să incrementăm indexul cu orice altă valoare in afară de unu, chiar și cu numere negative. Se numește +LOOP, și funcționează la fel.

```
: TEST1 10 0 DO I . 2 +LOOP ;<cr> Ok
TEST1<cr> 0 2 4 6 8 10 Ok
: TEST2 0 10 DO I . -2 +LOOP ;<cr> Ok
TEST2<cr> 10 8 6 4 2 0k
```

Observați cum +LOOP, ca și LOOP, nu depășește niciodată limita, chiar atunci cind merge in jos.

In legătură cu aceste bucle există un cuvint numit LEAVE (părăsește), care incheie o buclă la imediata intilnire a unui LOOP sau +LOOP. El face aceasta pur și simplu făcind indexul egal cu limita, astfel că la următoarea incrementare se depășește limita și bucla este părăsită.

```
: TEST3   -10 10   DO I 0<
   IF LEAVE ELSE I . THEN  -3   +LOOP ;<cr> Ok
TEST3<cr> 10 7 4 1 Ok
```

In acest exemplu am dorit să părăsim bucla cind indexul a devenit negativ, de aceea a apărut IF. Pentru că tot a venit vorba, o precizare: structurile de control pot fi incluse una intr-alta, dar avind grijă ca ele să nu se acopere parțial. De exemplu, ceva de genul

```
... DO ... IF ... LOOP ... THEN ...
```

este greșit și va fi semnalat ca eroare. Structura DO-LOOP trebuie să cuprindă complet pe IF-THEN astfel

```
... DO ... IF ... THEN ... LOOP ...
```

sau

```
... IF ... DO ... LOOP ... THEN ...
```

Am ajuns acum la ciclurile cu un număr necunoscut de repetări. Acestea au două forme

```
BEGIN acțiuni flag UNTIL
```

și

```
BEGIN acțiuni1 flag WHILE acțiuni2 REPEAT
```

In prima formă, secvența "acțiuni flag" se execută pină cind (until = pină cind) valoarea logică produsă de "flag" este adevărată. Este evident că bucla va fi parcursă cel puțin odată. Dacă acest lucru nu este de dorit, se folosește a doua formă. Aici, după executarea lui "acțiuni1" și producerea unui "flag", se continuă cu execuția lui "acțiuni2" atit timp cit (while = atit timp cit) valoarea logică este adevărată. După REPEAT controlul este trecut imediat după BEGIN la "acțiuni1".

Comparați aceste două definiții

```
: COUNT1   BEGIN DUP .  1- DUP 0= UNTIL DROP ;
: COUNT2   BEGIN DUP WHILE DUP . 1- REPEAT DROP ;
```

Ele fac același lucru. Așteaptă un număr in stivă și numără in jos spre zero. Dar dacă lui COUNT1 ii transmiteți un zero, acesta va număra mai departe sub zero, ceea ce in cazul lui COUNT2 nu se intimplă. Observați folosirea lui DUP pentru a transmite valoarea curentă de la o iterație la alta, și DROP-ul de la sfirșit pentru a elimina această valoare.

5. Variabile, constante și tablouri

Pină acum am folosit doar stiva pentru a păstra diverse rezultate. De multe ori este insă necesară folosirea variabilelor (variables). Acestea sint locații de memorie, cărora li se dă un nume convenabil, și sint folosite pentru păstrarea unor valori mai mult sau mai puțin temporare.

Să presupunem că programăm un joc pentru microcalculator in FORTH. Avem nevoie de o metodă de a păstra scorul curent. O variabilă este exact ceea ce ne trebuie.

```
0 VARIABLE SCORE
```

Variabila o numim SCORE, astfel incit să știm la ce folosește, și o inițializăm cu zero. Scorul curent il putem examina folosind "@" (se citește "fetch" și inseamnă "adu").

```
: TEST-SCORE   SCORE @ 0=
   IF ." Nimic marcat incă " THEN ;
TEST-SCORE<cr> Nimic marcat incă Ok
```

Putem inițializa variabilele cu "!" (citit "store" și insemnind "memorează").

```
5 SCORE !<cr> Ok
SCORE @ .<cr> 5 Ok
```

Pentru a examina conținutul unei variabile este comod să folosim "?" ("question" = intrebare). El este definit simplu ca

```
: ?   @ . ;
```

Astfel putem scrie

```
SCORE ?<cr> 5 Ok
```

Un operator interesant este "+!" ("plus-store"). El se foloseşte pentru a incrementa conţinutul unei variabile.

```
10 SCORE +!<cr>Ok
SCORE ?<cr> 15 Ok
-3 SCORE +!<cr> Ok
SCORE ?<cr> 12 Ok
```

Să vedem cum funcţionează toate acestea. Cînd spuneţi

```
0 VARIABLE SCORE
```

VARIABLE, care este un cuvint în dicţionar, creează un nou cuvînt - SCORE - şi face ca atunci cînd este invocat, SCORE să pună în stivă o adresă - adresa la care se memorează valoarea variabilei. Astfel, "!" aşteaptă o adresă în vîrful stivei şi o valoare dedesubt; el memorează valoarea la adresa indicată. "@" are nevoie de o adresă; el ia valoarea de la adresa respectivă şi o pune în stivă. Pe "+!" putem să-l definim noi înşine astfel

```
: +!   ( n adr -- )   SWAP OVER @ + SWAP ! ;
```

deşi el este definit în limbaj de asamblare pentru a rula cu viteză maximă.

Legat de variabile, este important să ştiţi că în sistem există una care se numeşte BASE şi care indică baza curentă de numeraţie. Două cuvinte pot modifica baza de numeraţie: DECIMAL, care face ca baza curentă să fie zece (acesta este cazul normal) şi HEX, care face ca baza să fie şaisprezece. Ele sînt definite astfel

```
: DECIMAL   10 BASE ! ;
: HEX   16 BASE ! ;
```

Putem vedea astfel cum arată diverse numere în cele două baze de numeraţie. Mai important chiar, putem vedea adresa la care este memorată o variabilă.

```
26 HEX .<cr> 1A Ok
OFF DECIMAL .<cr> 255 Ok
SCORE HEX . DECIMAL<cr> 4AE0 Ok
```

Observaţi cum în ultimul exemplu se readuce baza la zece după ce s-a tipărit valoarea adresei în hex. Cu BASE ne putem juca şi noi. Să scriem un cuvînt care să tipărească un număr în trei baze de numeraţie - zecimal, binar şi hexazecimal. Pentru aceasta definim întîi

```
: BINARY   2 BASE ! ;
: TRIBASE   DECIMAL DUP .
            HEX DUP . BINARY . DECIMAL ;
```

DUP-urile le folosim pentru a păstra numărul în stivă. La sfîrşit avem grijă să revenim la zecimal. Iată cum lucrează TRIBASE

```
7 TRIBASE<cr> 7 7 111 Ok
10 TRIBASE<cr> 10 A 1010 Ok
```

Probabil aţi înţeles că "@" şi "!" lucrează cu valori de 16 biţi. Uneori avem nevoie de valori mai mici, care încap pe numai 8 biţi. De exemplu, dacă scriem un editor de texte în FORTH, avem nevoie să examinăm caracterul curent pe care se află cursorul. Caracterul ocupă numai un octet, deci nu putem folosi "@". Pentru această situaţie există "C@". De asemenea, pentru memorarea unei valori de numai 8 biţi la o adresă, se foloseşte "C!" în loc de "!". În aceste două cuvinte, "C" vine de la "caracter", deoarece codurile caracterelor au nevoie de un singur octet pentru reprezentare.

Constantele se definesc astfel

```
250 CONSTANT BONUS
```

Cuvîntul CONSTANT creează, ca şi VARIABLE, un nou cuvînt în dicţionar - BONUS. Diferenţa este că acum cînd invocăm pe BONUS nu obţinem adresa la care este memorată valoarea, ci chiar valoarea.

```
BONUS .<cr> 250 Ok
```

Constantele se folosesc în primul rînd pentru a ne face viaţa mai uşoară. Dînd nume diverselor numere ne reamintim ce înseamnă ele. Deasemenea, modificările sînt simplu de efectuat dacă ceva se schimbă în program. În sfîrşit, facem puţină economie de memorie dacă un număr des folosit îl transformăm într-o constantă.

BONUS (= primă) de mai sus îl putem folosi într-un Joc pentru a denumi valoarea scorului de la care Jucătorul are dreptul la încă un Joc. Această botezare a unui număr este foarte utilă cînd numărul trebuie, din diverse motive, schimbat (de exemplu Jocul devine foarte dificil şi trebuie să reducem valoarea de la 250 la 200). Tot ce trebuie să facem este să schimbăm definiţia lui BONUS. Dacă am fi folosit numărul 250 explicit, ar fi trebuit să căutăm în tot listing-ul şi să înlocuim toate apariţiile sale.

Pentru a discuta despre tablouri, să presupunem că în Jocul pe care tocmai îl programăm pot Juca pînă la cinci Jucători simultan. Dorim ca scorurile Jucătorilor să fie tratate într-o manieră unitară. Putem folosi atunci un vector cu cinci elemente, care să păstreze scorurile. În FORTH aceasta se scrie astfel

```
0 VARIABLE SCORES  8 ALLOT
```

Am creat o variabilă SCORES, rezervînd astfel 2 octeţi pentru valoarea ei. Fraza "8 ALLOT" rezervă un spaţiu suplimentar de 8 octeţi (allot = alocă), astfel că pentru SCORES există acum 2 + 8 = 10 octeţi, cîte doi pentru fiecare Jucător.

Dorim să facem ca toate scorurile iniţiale să fie zero. Pentru aceasta folosim

```
SCORES 10 0 FILL
```

Cînd invocăm pe SCORES, acesta ne furnizează adresa primului din cei zece octeţi rezervaţi. Cuvîntul FILL are nevoie de trei argumente: o adresă, un număr de octeţi şi o valoare cu care va umple (fill = umple) respectivii octeţi de la adresa furnizată.

Există două valori particulare cu care dorim să umplem o succesiune de octeţi. Una este zero, ca mai sus, şi cealaltă este codul pentru caracterul "spaţiu", numit şi "blank". De aceea există două cuvinte asemănătoare cu FILL, dar care nu mai au

nevoie decît de adresă şi număr de octeţi. Ele sînt ERASE şi BLANKS. De exemplu, fraza de mai sus putea fi scrisă

```
SCORES 10 ERASE
```

Dacă doream să umplem cu spaţii, am fi scris

```
SCORES 10 BLANKS
```

Cînd vrem să avem acces la unul din cele cinci scoruri, de exemplu la al doilea, scriem

```
SCORES 2+
```

dar aceasta este mai anevoios. Creeăm atunci un cuvînt căruia i se furnizează un număr de jucător, între zero şi patru, şi care ne furnizează adresa unde se află scorul respectivului jucător

```
: PLAYER   ( player# -- 'score)   2* SCORES + ;
```

Vedeţi cum în comentariu am notat ceea ce intră (player# = player number = numărul jucătorului) şi ceea ce rezultă ('score = address of score = adresa scorului); acestea sînt convenţii standard pentru comentarea efectelor asupra stivei. "2*" se foloseşte pentru că fiecărui jucător i s-au alocat doi octeţi; obţinem astfel un **offset** în tablou. "SCORES +" adună offsetul la adresa de început a tabloului, rezultînd astfel adresa elementului.

Dacă nu vă place să număraţi jucătorii de la zero la patru, puteţi să-i număraţi şi de la unu la cinci, dar atunci trebuie să-l schimbaţi pe PLAYER în

```
: PLAYER   1- 2* SCORES + ;
```

Vedeţi deci că este mai convenabil să începeţi cu zero.

Pe PLAYER îl putem folosi astfel (considerînd varianta cu număratul de la zero)

```
25  3 PLAYER  !
```

Aceasta s-ar citi cam aşa: pune scorul jucătorului numărul patru (3 + 1 = 4) la 25. Pentru a mări scorul jucătorului numărul doi cu 35 de puncte scriem

```
35  1 PLAYER  +!
```

Probabil că cei care cunosc BASIC sau alte limbaje de programare tradiţionale sînt surprinşi de modul în care FORTH tratează vectorii. Trebuie să scriem noi înşine mecanismul prin care facem indexarea. Dar acest lucru ne asigură o mare flexibilitate în modul de proiectare al programului. De exemplu, în PLAYER putem introduce cuvinte care să verifice că i se furnizează un număr de jucător valid, între zero şi patru. Odată programul testat, putem elimina aceste verificări din varianta finală, căci ele consumă timp.

FORTH nu doreşte să mulţumească pe toată lumea asigurînd tot felul de construcţii exotice. El nu are o instrucţiune CASE cum există în Pascal. Motivul este că în majoritatea cazurilor nu este nevoie de ea, iar dacă o doriţi totuşi neapărat, puteţi scrie singuri una, nu este deloc complicat. FORTH asigură mijloace prin care puteţi extinde compilatorul însuşi, ceea ce pentru limbajele tradiţionale este aproape imposibil.

6. Din nou despre numere

Numerele de lungime dublă ocupă 32 de biţi (două celule) şi pot avea valori între +2147483647 şi -2147483648 (!).

NUMBER ştie că este vorba despre un număr de lungime dublă dacă acesta conţine un punct. Astfel de numere pot fi afişate cu "D." ("D-dot").

```
10.0<cr> Ok
D.<cr> 100 Ok
```

Punctul nu înseamnă decît că numărul este pe 32 de biţi. Astfel 1.00, 100. şi .100 se reprezintă toate la fel. Singura diferenţă este în valoarea variabilei DPL care indică poziţia celui mai din dreapta punct. Dacă DPL este negativ, înseamnă că a fost vorba despre un număr pe 16 biţi.

```
100<cr> Ok
DPL ?<cr> -1 Ok
100. D.<cr> 100 Ok
DPL ?<cr> 0 Ok
10.0 D.<cr> 100 Ok
DPL ?<cr> 1 Ok
```

În stivă un număr pe 32 de biţi se reprezintă cu cei mai semnificativi 16 biţi în vîrf şi cu ceilalţi 16 dedesubt. În exemplul următor nu uitaţi că dot nu afişează decît numere pe 16 biţi.

```
100.<cr> Ok
. .<cr> 0 100 Ok
```

Pentru că o sută încape pe 16 biţi, partea lui cea mai semnificativă este zero şi aceasta este valoarea afişată de primul dot.

Manevrarea numerelor de lungime dublă se face cu operatori speciali. Deoarece ei sînt similari cu cei pentru numere de lungime simplă, îi voi lista doar, fără a mai da exemple de utilizare.

2SWAP	(d1 d2 -- d2 d1) Inversează primele două elemente din stivă	"2-swap"
2DUP	(d -- d d) Duplică vîrful stivei	"2-dupe"
2OVER	(d1 d2 -- d1 d2 d1) Copiază al doilea element şi îl pune în vîrf	"2-over"
2ROT	(d1 d2 d3 -- d2 d3 d1) Roteşte al treilea element în vîrf	"2-rote"
2DROP	(d2 --) "Aruncă" elementul din vîrf	"2-drop"
D+	(d1 d2 -- d-sum) Adună	"d-plus"
DMINUS	(d -- -d) Schimbă semnul	"d-minus"
DABS	(d -- |d|) Valoarea absolută	"d-abs"

Există şi variabile şi constante pe lungime dublă. Acestea se declară cu 2VARIABLE şi 2CONSTANT. Echivalenţii lui fetch şi store sînt 2@ şi 2!.

Numerele fără semn sînt întotdeauna pozitive. Dacă sînt pe 16 biţi, atunci ele au valori între 0 şi 65535; pe 32 de biţi ele merg de la 0 la 4294967295.

NUMBER (cel care se ocupă de numere cînd acestea sînt introduse în şirul de intrare) nu face nici-un fel de verificare a valorilor. Aceasta înseamnă că se pot introduce numere de orice lungime, dar nu se vor păstra decît cei mai

puţini semnificativi 16 sau 32 de biţi (după cum numărul conţine sau nu punctul zecimal). Deasemenea, NUMBER poate trata atit numere cu semn, cit şi fără semn.

De exemplu, dacă se introduce orice număr intre 32768 şi 65535, NUMBER il va converti intr-un număr fără semn. Orice valoare intre -32768 şi -1 va fi memorată ca un număr cu semn in cod complementar.

Toate acestea inseamnă că stiva poate fi folosită pentru a păstra fie numere cu semn, fie numere fără semn. Incadrarea valorilor binare in una din cele două categorii depinde de operatorii ce le sint aplicaţi. Este sarcina programatorului de a stabili ce formă este potrivită unei anumite situaţii şi de a folosi operatorii corespunzători.

Cuvintul "." (dot) tipăreşte valoarea unui număr cu semn:

```
65535 .<cr> -1 Ok
```

Pentru a obţine o valoare fără semn se foloseşte U. (u-dot):

```
65535 u.<cr> 65535 Ok
```

In tabelul următor se prezintă alţi operatori care lucrează cu numere fără semn; "u" reprezintă o valoare pe 16 biţi, iar "ud" o valoare pe 32 de biţi.

```
U*   ( u1 u2 -- ud )   ud = u1 * u2   "u-star"
U/   ( ud u1 -- u2 u3 )               "u-slash"
     Imparte ud la u1 obţinind
     restul u2 şi citul u3
U<   ( u1 u2 -- f )                   "u-mai-mic"
     f = adevărat dacă u1 < u2
```

In incheierea acestei secţiuni vom discuta despre metode de afişare formatată a numerelor. Cea mai simplă definiţie de acest tip este următoarea

```
: UD.  <#  #s  #>  type space ;
```

UD. va tipări un număr fără semn de lungime dublă. Cuvintele "<#" şi "#>" (pronunţate "bracket-number" şi "number-bracket") specifică inceputul şi sfirşitul procesului de conversie. In această definiţie, conversia este executată de cuvinul "#s" (pronunţat "numbers"). "#s" converteşte o valoare din stivă in caractere ASCII. El va produce atitea cifre cite sint necesare pentru a reprezenta numărul. Cuvintul TYPE va afişa caracterele ce reprezintă numărul.

```
12.345 ud.<cr> 12345 Ok
12. ud.<cr> 12 Ok
0. ud.<cr> 0 Ok
```

Să presupunem că avem nevoie de un cuvint care să afişeze date. Considerind că in stivă se află numărul fără semn de lungime dublă 31071962, dorim să-l afişăm ca 31-07-1962 (adică 31 iulie 1962). Pentru aceasta definim

```
: .DATE  <#  # # # #  ascii - hold
  # #  ascii - hold  #s  #>  type space ;
```

Cuvintul "#" (pronunţat "number") produce numai o singură cifră. Ordinea conversiei este inversă faţă de cea in care apare numărul, astfel că fraza "# # # #" produce cele mai din dreapta patru cifre, care ne vor da anul.

"ascii -" va produce codul ASCII al liniuţei de separaţie, iar "hold" va pune acest cod după primele cifre (hold = păstrează). "# #" produce cele două cifre ale lunii, după care urmează din nou liniuţa de separaţie. In final, "#s" converteşte restul numărului. Procesul este incheiat de "#>" iar TYPE afişează rezultatul.

```
31.07.1962 .date<cr> 31-07-1962 Ok
9.05.1877 .date<cr> 9-05-1877 Ok
```

7. In loc de incheiere

Am ajuns la sfirşitul lecţiei noastre introductive despre FORTH. Nu am vorbit despre multe lucruri interesante ca: structura unui cuvint in dicţionar, extinderea compilatorului, vocabulare, folosirea asamblorului, etc.

Aceste lucruri nu sint foarte complicate, dar necesită puţină experienţă pentru a le inţelege. De asemenea, este nevoie de mult spaţiu pentru explicaţii. Oricum, acesta este un articol introductiv, şi scopul său este de a vă forma o idee, nu de a vă transforma in experţi.

FORTH este un limbaj puternic, dar deosebit de ceea ce exista pină la el, şi de aceea a fost intimpinat cu neincredere. Neincrederea s-a risipit insă in faţa posibilităţilor sale. El este folosit in aplicaţii foarte diverse, de la comanda unei aprinderi electronice pentru automobile, pină la controlul unui intreg observator astronomic.

FORTH asigură mijloacele prin care se poate maximiza eficienţa folosirii calculatorului. El este mai rapid decit limbajele tradiţionale. Faţă de programele echivalente in limbaj maşină este cu 20-75% mai lent, dar porţiunile critice pot fi codificate direct in limbaj de asamblare şi vor rula cu viteza maximă a procesorului.

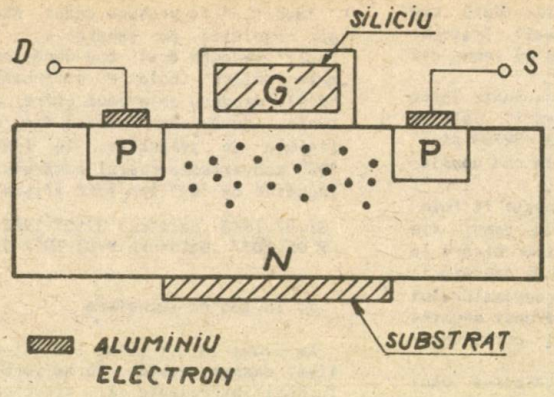
Codul compilat de FORTH este compact. Aplicaţiile scrise in FORTH necesită mai puţină memorie decit programele echivalente in limbaj de asamblare ! Scris in FORTH, intregul sistem de operare impreună cu setul standard de cuvinte ocupă mai puţin de 8 Kbytes.

Timpul necesar scrierii unei aplicaţii in FORTH este de zece ori mai mic faţă de timpul necesar scrierii in limbaj de asamblare, şi de două ori mai mic decit programarea in limbaje de nivel inalt. Aceasta pentru că FORTH foloseşte programarea structurată, pentru că este interactiv şi modular. In plus, toate uneltele necesare scrierii programelor - interpreter, compilator, asamblor, depanator, etc. - sint integrate intr-un sistem unic, permiţind un acces rapid.

Dacă doriţi să invăţaţi FORTH, incepeţi prin a lucra exemplele din acest articol. Recitiţi cu atenţie manualul FORTH-ului dumneavoastră - probabil că acum veţi inţelege mai multe lucruri. De asemenea, urmăriţi paginile revistei "Tehnium" in care vor apare exemple de programe. Pentru intrebări şi neclarităţi vă puteţi adresa in scris autorului pe adresa redacţiei.

Bibliografie

1. Leo Brodie, Starting FORTH, Prentice-Hall, Inc., Englewood Cliffs, 1982
2. Leo Brodie, Thinking FORTH, Prentice-Hall, Inc., Englewood Cliffs, 1984
3. Charles Eaker, X-FORTH User's Manual

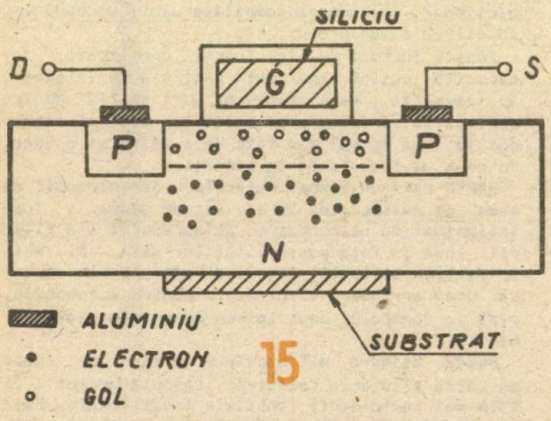


14

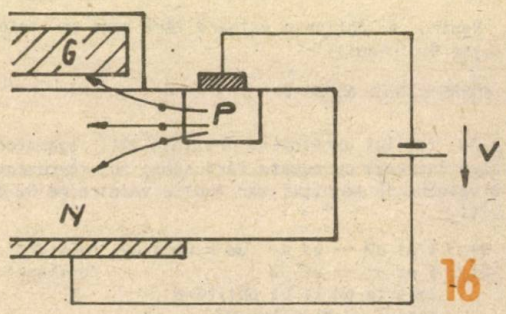
(URMARE DIN PAG. 70)

O memorie EPROM programată poate fi ștearsă în vederea unei noi programări. Pentru a șterge o memorie EPROM trebuie, printr-un procedeu oarecare, descărcată poarta tranzistorului MOS.

Un astfel de procedeu este următorul: o fereastră specială prevăzută în capsula ansamblului-memorie permite iradierea tuturor celulelor cu radiații U.V. (ultraviolete). Efectul constă în transferarea excesului de sarcini din grilă în substrat: în consecință, memoria este „ștearsă" și capabilă să suporte o nouă programare.



15



16

memorator

CONDENSATOARE ELECTROLITICE

Eg 7400

EG 74.73	220	63	450	0,54
EG 74.74	330	63	650	0,36
EG 74.75	470	63	770	0,25
EG 74.78	680	63	1050	0,17
EG 74.78	1000	63	1400	0,12
EG 74.78	1500	63	1950	0,10
EG 74.80	2200	63	1950	0,10
EG 74.73	100	100	350	1,00
EG 74.74	150	100	470	0,66
EG 74.76	220	100	560	0.45

(URMARE DIN PAG. 63)

HiFi

PROPRIETĂȚILE SUNETULUI

Ing. GH. GRIGORE

Oscilația corpurilor materiale se transmite mediului elastic înconjurător sub forma de unde sonore care radiază în spațiu în toate direcțiile. Ajunse la ureche, oscilațiile acustice, avînd o frecvență cuprinsă între 16—20 000 Hz, produc senzația de sunet.

Noțiunea de sunet are un dublu aspect; unul fizic (acustic) și unul psihofiziologic — subiectiv. Din punct de vedere fizic, sunetul este o vibrație acustică capabilă să producă o senzație auditivă (deci în limitele de frecvență și amplitudine perceptibile urechii omenești). Din punct de vedere subiectiv, sunetul este o senzație auditivă produsă de vibrația acustică, tot astfel precum lumina și culoarea sînt senzații vizuale produse de vibrațiile electromagnetice în limitele perceptibile văzului.

Din punct de vedere al compoziției spectrale și al vibrațiilor sonore conținute, sunetele se clasifică în sunete pure și sunete complexe (muzicale, acorduri, zgomote).

Sunetele pure sînt rezultatul unor oscilații acustice sinusoidale, neîntovărășite de sunete armonice. În natură, asemenea sunete sînt rare. Se poate considera că sunetul produs de un diapazon lovit ușor este un sunet pur, practic lipsit de armonici. De asemenea, sunetele sinusoidale pure mai pot fi produse și de generatoare de ton utilizate la măsurări și etalonări electroacustice, precum și de unele instrumente muzicale electronice.

În general, în natură sunetele sînt întovărășite de armonice, fiind deci sunete complexe. Se numește armonică o vibrație sinusoidală a cărei frecvență este un multiplu întreg al frecvenței fundamentale, iar un sunet complex sunetul care conține mai multe sunete pure. Instrumentele muzicale produc de obicei asemenea sunete complexe.

Din punct de vedere fizic, un sunet complex se caracterizează prin: FRECVENȚA celui mai intens sunet component pur, care de cele mai multe ori este sunetul fundamental; INTENSITATEA ACUSTICĂ; TIMBRUL (componenta spectrală); DURATĂ; REGIMUL TRANZITORIU (modul de stabilire și încetare a sunetului).

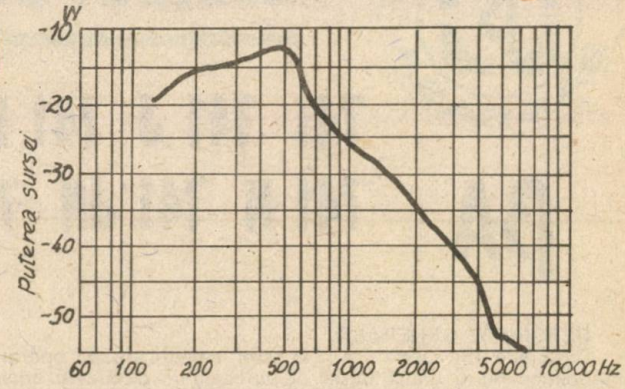
PROPAGAREA SUNETULUI ÎN AER LIBER. Sunetul radiat de o sursă sonoră se propagă în spațiu sub forma unor unde sonore. Similar cu undele de la suprafața apei, care iau forma unor cercuri cu centrul în punctul de oscilație și cu razele din ce în ce mai mari, undele sonore iau forma unor sfere, avînd în centru sursa care le produce. Aceste sfere se măresc din ce în ce mai mult și, cum undele sonore transportă o anumită energie, este ușor de observat că această energie se repartizează pe o suprafață din ce în ce mai mare.

Dacă în calea undelor se interpune un obstacol, acesta în general este ocolit de undele sonore, fenomenul purtînd denumirea de DIFRACȚIA SUNETULUI. Acest fenomen este cu atît mai evident cu cît dimensiunile obstacolului sînt mai mici în comparație cu lungimea de undă a sunetului. Dacă însă obiectul interpus în calea undelor sonore are dimensiuni mari, undele sonore sînt reflectate de acest obstacol întorcîndu-se spre sursă. În acest caz, o parte din energia sonoră incidentă se reflectă constituind energia undelor reflectante. O altă parte din energia incidentă pătrunde în corpul care constituie obstacolul.

Se constată că mărimea energiei reflectate depinde de natura suprafeței de separare a mediilor pe care se produce reflexia. De exemplu, dacă obstacolul plasat în calea undelor sonore este un perete de beton lustruit, aproape întreaga energie sonoră incidentă este reflectată.

Cu totul altfel se prezintă situația dacă suprafața peretelui

1



Caracteristica de frecvență a vocii omenești medii

este acoperită cu un strat de material absorbant ca, de exemplu, pîsla sau vata minerală. În acest caz, o parte din energia sonoră incidentă este absorbită. În funcție de calitatea și cantitatea materialului absorbant se poate ajunge pînă la absorbția totală a acestei energii cînd energia sonoră reflectată este nulă.

PROPAGAREA SUNETULUI ÎN SPAȚII ÎNCHISE. Într-o încăpere, undele produse de o sursă din interior se reflectă atunci cînd întîlnesc suprafețele delimitatoare (pereții, tavanul, pardoseala), întorcîndu-se din nou în încăpere. Pentru un ascultător prezent într-o încăpere întreaga energie sonoră care nu se întoarce în încăpere se consideră absorbită de suprafețele care delimitează încăperea. Se ajunge, astfel, la definirea coeficientului de absorbție sonoră (α), care este raportul dintre energia sonoră absorbită de o suprafață și energia sonoră incidentă pe acea suprafață.

Absorția energiei sonore se datorează, pe de o parte, disipației de energie în materialul din care este construită suprafața unui perete, tavan sau pardoseală (tencuială, draperii, covoare etc.) și pe altă parte transmisiei energiei prin ecrane despărțitoare (pereți, planșee). Pentru a se caracteriza gradul de disipație a energiei sonore și gradul de transmisie sonoră s-au definit doi coeficienți, și anume: coeficientul de disipație sonoră (δ), care este raportul dintre energia sonoră disipată în material și energia sonoră incidentă pe suprafața materialului respectiv, și coeficientul de transmisie sonoră (τ), care este raportul dintre energia sonoră incidentă pe suprafața ecranului despărțitor.

Față de cele arătate mai sus, între acești doi coeficienți și coeficientul de absorbție sonoră există relația:

$$\alpha = \delta + \tau$$

Intreaga energie sonora care nu este absorbită se reflectă prin coeficientul de reflexie (ψ), definindu-se raportul dintre energia sonoră reflectată și energia sonoră incidentă.

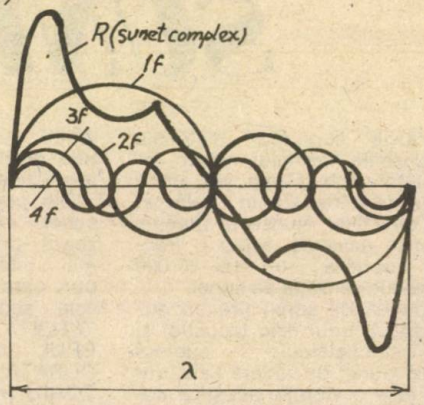
Între coeficientul de absorbție sonoră și coeficientul de reflexie există relația:

$$\alpha + \psi = 1$$

Ca urmare, un ascultător plasat în interiorul unei încăperi aude atît sunetul direct provenit de la sursa sonoră, cît și o suma de sunete reflectate, parvenite prin reflexie pe pereți, tavan și pardoseală. Datorită faptului ca la fiecare reflexie aceste suprafețe absorb o anumită cantitate de energie sonoră, valoarea intensității sunetului reflectat depinde nu numai de puterea acustică a sursei sonore, ci și de

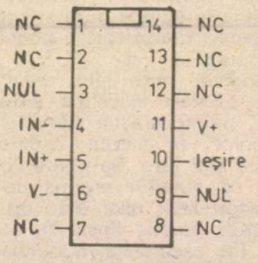
Descompunerea unui sunet complet R în armonici (λ = lungimea de undă)

2



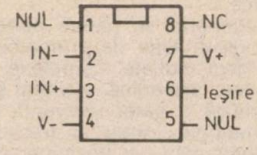
memorator

741 741 J 741 M
βA 741 N 741 JN 741 MN



βA 741/741J / 741M

Configurația terminalelor



βA 741N/741JN/741MN

DESCRIERE GENERALĂ

Circuitele din seria βA 741 sînt amplificatoare operaționale monolitice cu o gamă largă de aplicații în circuitele analogice. Acestea se caracterizează printr-o gamă largă pentru tensiunile de intrare, protecție internă la „agățare" (latch-up), protecție internă la scurtcircuit, compensare cu frecvența.

absorbția din încăpere.

În cazul încăperilor mici (locuințe), contează numai sunetul direct, fiind neglijat de data aceasta sunetul reflectat. Avînd în vedere că în aceste cazuri distanțele sînt mici, nu sînt necesare puteri instalate mari.

Raportul dintre intensitatea sunetului direct și intensitatea sunetului reflectat are mare importanță în cazul captării sunetului cu microfoane. Dacă microfonul se plasează prea aproape de sursa sonoră, de exemplu, un vorbitor sau un solist, sunetul apare mat, lipsit de culoare, parcă s-ar produce în aer liber; acest lucru este datorat preponderenței pe care o au sunetele directe.

În plus, cele mai slabe zgomote, precum și respirația vorbitorului sînt captate și la redare apar amplificate în proporții exagerate. Dacă microfonul este plasat la o depărtare prea mare de sursă, imaginea sonoră apare slab conturată, neclară, parcă sursa s-ar pierde în depărtare.

Între aceste două situații extreme există o distanță optimă între sursă și microfon și, în practică, s-a constatat că această distanță este tocmai aceea la care intensitatea sunetului direct devine egală cu intensitatea sunetului reflectat.

Pînă acum am considerat că sursa sonoră emite un sunet continuu, adică ne găsim în regim permanent. Dacă însă la un moment dat sursa sonoră încetează radiația, sunetul produs va mai persista un timp în încăpere, după care se va stinge complet. Într-adevăr, sunetele sînt reflectate de suprafețele care limitează încăperea și, la fiecare reflexie, o parte din energia sonoră este absorbită, așa încît descreșterea intensității sunetului se va produce în mod treptat, pînă în momentul în care energia sonoră rămasă este prea mică pentru a mai produce o senzație auditivă.

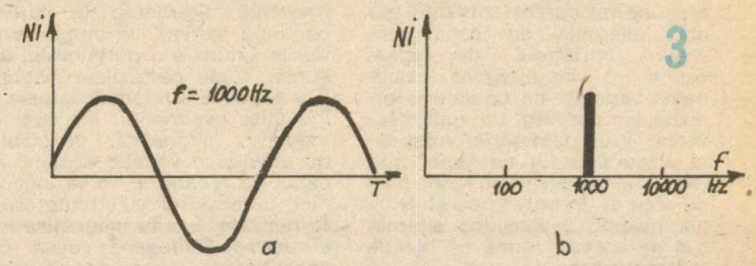
Această prelungire a sunetului într-o încăpere, după încetarea radiației sonore, se numește REVERBERAȚIE. Aceasta este cu atît mai mare cu cît volumul sălii este mai mare și cu cît absorbția corespunzătoare este mai mică. Un exemplu tipic de încăperi care au această caracteristică pregnantă sînt bisericile mari și catedralele în care se cîntă curent la orgă. Unele au calități așa de bune încît s-au putut face în ele înregistrări excelente.

Nu trebuie confundată noțiunea de reverberație cu cea de ecou. În timp ce ecoul este o reproducere distinctă a unui sunet, ca urmare a reflexiei undelor sonore de către un ecran reflectat interpus în calea propagării lor, reverberația este o prelungire a sunetului datorită numeroaselor reflexii care se produc pe toate suprafețele interioare ale unei încăperi. De aceea este greșit a se spune despre o sală reverberantă că este o sală cu ecou. Ecoul se poate produce și într-o sală cu reverberație redusă, dacă sînt îndeplinite condițiile necesare pentru producerea lui. De asemenea, ecoul se produce și în aer liber, unde nu poate fi vorba de reverberație.

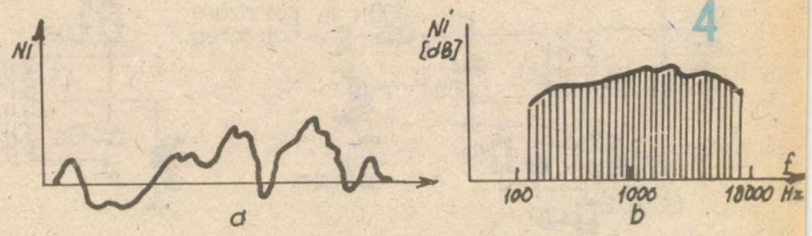
Cercetările experimentale au arătat că durata persistenței unei senzații auditive, din momentul întreruperii ei, este de aproximativ 1/20 s sau 50 ms.

Pentru a se produce un ecou, intervalul de timp care se scurge din momentul recepționării sunetului direct pînă în momentul recepționării celui reflectat trebuie să fie mai mare decît 50 ms. Această condiție se mai poate formula și prin diferența dintre drumul parcurs de sunetul direct și cel reflectat, care trebuie să fie mai mare de 17 m. În plus, pentru a se putea recepționa un ecou bine precizat, este necesar ca ecranul interpus în calea undelor să fie puternic reflectant, pentru ca nivelul de intensitate sonoră al sunetului reflectat să fie apropiat de nivelul de intensitate sonoră a sunetului direct.

Dacă se produce un sunet într-o sală în care există doi pereți paraleli și destul de reflectanți, acesta se va reflecta succesiv de fiecare perete aproape fără să se piardă din intensitate. Se produc astfel o serie de ecouri care se succed la intervale de timp foarte scurte unul după altul și a căror intensitate scade lent.



Reprezentarea unui sunet pur (sinusoidal).
a - oscilograma; b - spectrul sonor.



Reprezentarea unui zgomot
a - oscilograma; b - spectrul sonor.

Acest fenomen este cunoscut sub denumirea de ecou de fluture și producerea lui creează o senzație auditivă neplăcută. La sălile care au acest defect, pentru a pune în evidență ecoul de fluture, se bate o dată din palme, auzindu-se apoi un sunet metalic, similar cu zăngănitul unei table.

7

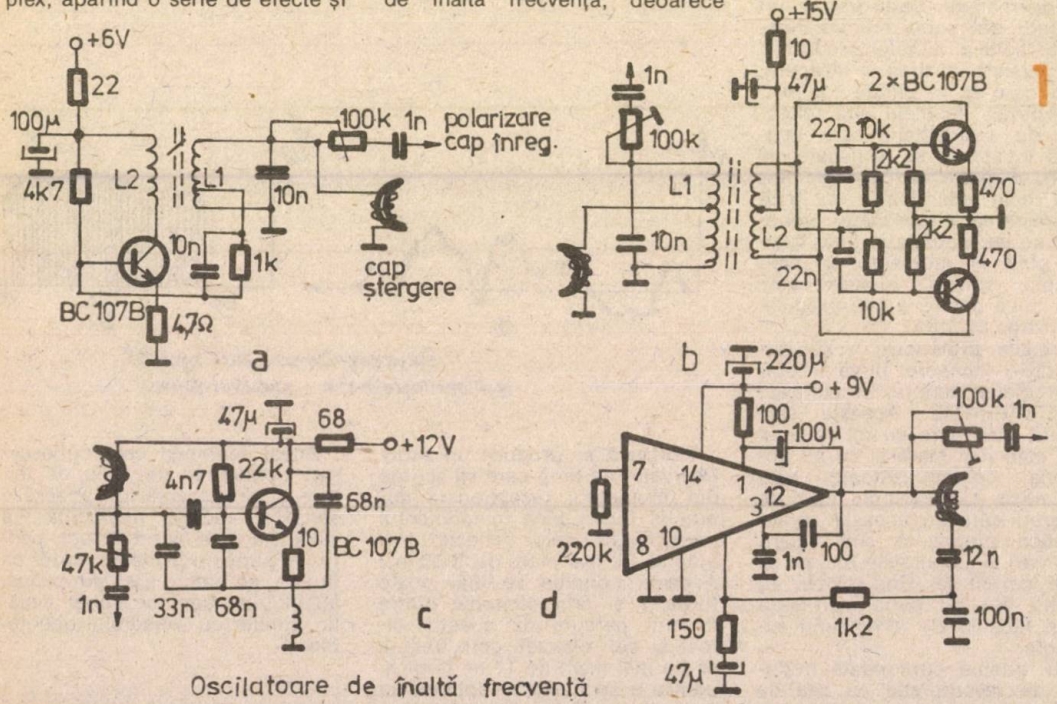
ÎNREGISTRAREA MAGNETICĂ A SUNETULUI

DAN BĂLĂNESCU, Cluj-Napoca

Înregistrarea magnetică a sunetului se bazează pe magnetizarea variabilă a unui purtător care se deplasează prin fața unui electromagnet (cap de înregistrare) prin bobina căruia circulă un curent alternativ de audiofrecvență peste care se suprapune un curent continuu sau unul alternativ de înaltă frecvență. Purtătorul de sunet capătă o magnetizare remanentă variabilă ce conține informația din curentul de audiofrecvență. Prin deplasarea ulterioară a purtătorului de sunet prin fața unui alt electromagnet (cap de redare), în bobina acestuia va lua naștere o tensiune alternativă de aceeași formă cu cea de la înregistrare.

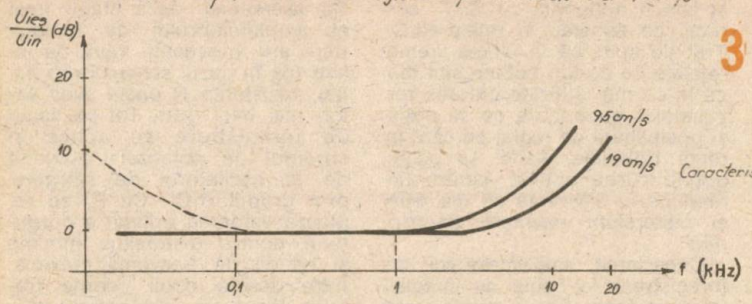
În realitate, procesul înregistrării magnetice este mai complex, apărînd o serie de efecte și dificultăți ce trebuie înlăturate. El comportă 3 faze: ștergerea, înregistrarea și redarea.

Ștergerea este operația ce se efectuează pentru anularea înregistrării anterioare. Se poate face folosind curent continuu sau curent alternativ de înaltă frecvență. Ștergerea în curent continuu constă în magnetizarea la saturație a purtătorului de sunet. Toate particulele magnetice vor avea o anumită valoare a inducției remanente, iar cum o inducție magnetică constantă nu creează o variație de flux, în capul de redare nu se va induce nici o tensiune electromotoare. În realitate, banda magnetică nu e perfect omogenă, ceea ce constituie o sursă de zgomot. Acest neajuns se înlătură în cazul ștergerii în curent alternativ de înaltă frecvență, deoarece prin acest procedeu banda e demagnetizată total. Este deci evidentă superioritatea acestei metode. Este important ca nivelul curentului de ștergere să fie suficient de mare pentru ca ștergerea să fie completă. El depinde de tipul benzii folosite și e proporțional cu curentul de premagnetizare. În figura 1 sînt date cîteva scheme de oscilatoare de înaltă frecvență. Oscilatoarele din figurile 1 a, b folosesc bobine construite pe carcase tip oală din ferită. Pentru prima schemă L1 are circa 100 spire (priză pe la spira 30) și L2 are 15 spire. La schema a doua, înfășurarea acordată are 150 spire (priză la spira 100) și înfășurările de reacție au cîte 30 spire. În funcție de condensatoarele de acord, se poate modi-



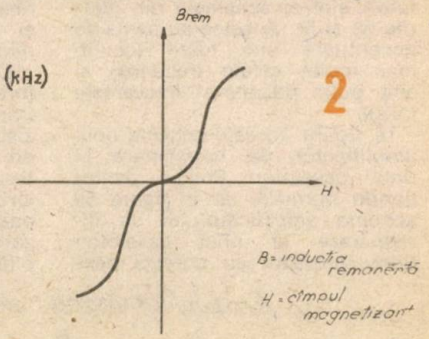
Oscilatoare de înaltă frecvență

Caracteristica amplificatorului de înregistrare pentru vitezele 19 și 9,5 cm/s



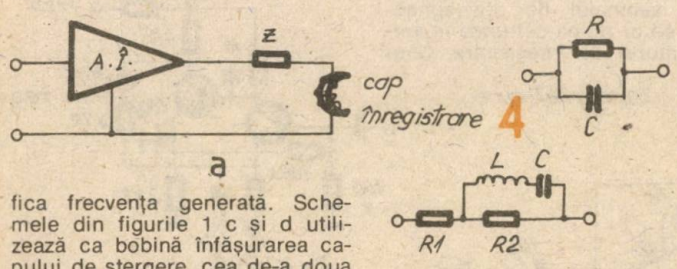
3

Caracteristica de transfer a purtătorului de sunet



2

Conectarea rețelei de corecție în serie cu capul de înregistrare



4

fica frecvența generată. Schemele din figurile 1 c și d utilizează ca bobină înfășurarea capului de ștergere, cea de-a doua fiind utilă datorită folosirii circuitului integrat TBA 790 ce poate fi folosit la vedere pentru amplificator de audiofrecvență.

Înregistrarea este procesul cel mai complex, de care depind în cea mai mare măsură calitățile obținute. Cum caracteristica de transfer a benzii (caracteristica ce exprimă inducția remanentă în funcție de cîmpul magnetizant) nu este liniară (fig. 2), semnalul înregistrat va avea distorsiuni. De aceea, se caută ca semnalul de audiofrecvență să fie plasat într-o zonă liniară a acestei caracteristici, ceea ce se face prin polarizarea capului de înregistrare, folosind curent continuu (se utilizează o singură porțiune liniară a caracteristicii: cea de sus sau cea de jos) sau folosind curent alternativ (se utilizează ambele porțiuni liniare). Polarizarea în curent alternativ oferă rezultate superioare prin nivelul zgomotului de fond mai redus și distorsiunile scăzute. Trebuie pusă condiția ca frecvența curentului alternativ de polarizare să fie de cel puțin 5 ori mai mare ca frecvența cea mai înaltă înregistrată, iar nivelul lui să fie astfel ales încît semnalul util să nu depășească marginile liniare ale caracteristicii. De asemenea, nivelul inducției remanente în bandă nu este constant în funcție de frecvență, el scade cu cît frecvența crește, de aceea, la înregistrare e necesară o ridicare suplimentară a frecvențelor înalte, ridicare ce trebuie să fie cu atît mai mare cu cît viteza e mai mică. Dar la trecerea peste un anumit nivel al semnalului util în capul de înregistrare se pot depăși marginile liniare ale curbei, ceea ce va duce evident la distorsiuni, fără ca nivelul semnalului util din bandă (inducția magnetică remanentă) să aibă o creștere sensibilă. De aici rezultă o limitare a benzii superioare de frecvență ce poate fi înregistrată pentru o anumită viteză și un anumit cap de înregistrare. Frecvența înregistrată poate fi cu atît mai mare cu cît viteza e mai mare și cu cît întrefierul capului de înregistrare e mai mic.

Corecțiile aplicate la înregistrare nu sînt standardizate (panta curbei, accentuarea maximă), ele depind de banda folosită și de capul de înregistrare. În general, forma curbei la înregistrare e cea din figura 3. Nivelul maxim de accentuare nu trebuie să depășească 25 dB (față de nivelul de la 1 000 Hz). Uneori, pentru a obține la redare o bandă cît mai liniară, se ridică cu 5—10 dB și frecvențele joase, sub 100 Hz (în special la viteze mai mari ale benzii).

Nivelul curentului de premagnetizare, considerat 100% în cazul benzilor normale (acoperite cu oxizi de fier), depinde de tipul benzii utilizate conform tabelului următor.

FeO	100%
FeCr	110%
CrO_2	140%
Metal	250%

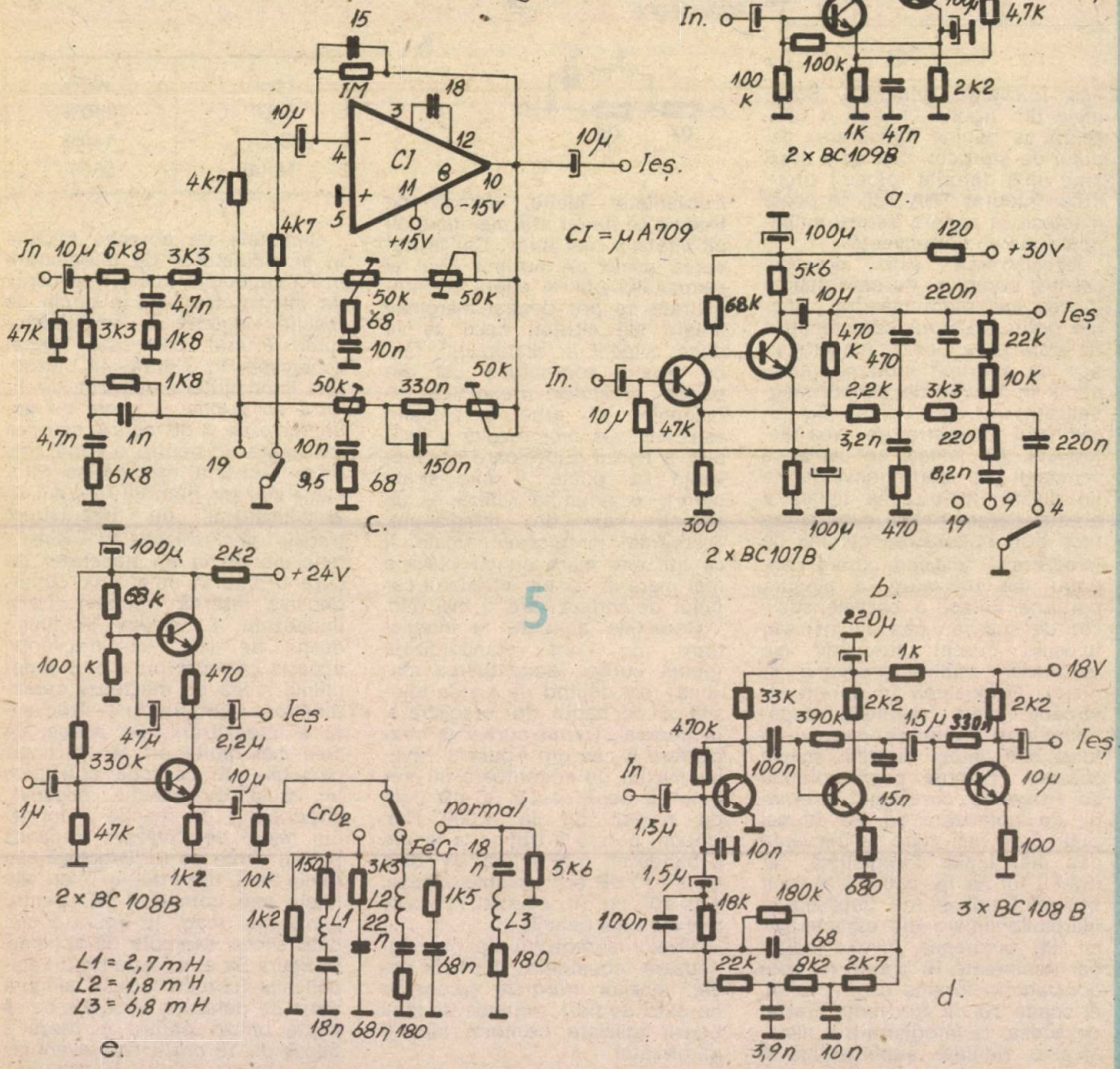
Circuitele de corecție folosite în amplificatorul de înregistrare pot fi conectate în serie cu capul de înregistrare sau în bucla de reacție negativă a amplificatorului. Primul caz este figurat schematic în figura 4a. Structura impedanței z este cea din figura 4b pentru o pantă de accentuare de 6 dB/octavă sau cea din figura 4c pentru o pantă mai mare. Această metodă se folosește mai rar datorită faptului că amplificatorul de înregistrare trebuie să furnizeze la ieșire o tensiune de la un generator de curent (o sursă de semnal cu impedanță internă ridicată). Dacă impedanța 7 variază în limite destul de largi cu frecvența, această condiție nu e bine îndeplinită, ceea ce afectează caracteristica amplitudine—frecvență a înregistrării. De aceea, de cele mai multe ori se utilizează circuitele de corecție conectate în bucla de reacție negativă. Cvadripolul de reacție e format din rețele de defazare trece-jos pentru pante de 6 dB/octavă sau rețele RLC, filtre dublu T, ori alte rețele mai complexe RC pentru pante mai mari. În figura 5 sînt date cîteva exemple de scheme. În figura 5a e reprezentat un amplificator corector cu o singură rețea de defazare trece-jos ce se poate utiliza pentru o viteză a benzii de 19 cm/s, sau, eventual

prin mărirea condensatorului C și pe 9,5 cm/s. Tot pentru magnetofon sînt și schemele din figurile 5b și 5c. Acestea au panta de accentuare mai mare (conțin mai multe celule trece-jos) și sînt puțin ridicate și frecvențele joase.

În figura 5d este schema unui amplificator de înregistrare al unui casetofon Philips pentru banda normală, iar în figura 5e schema amplificatorului de înregistrare al unui casetofon Sony prevăzut cu corecții pentru bandă normală, bandă FeCr și bandă acoperită cu CrO_2, corecții ce constau în rețele RLC. Trebuie spus că în ultima vreme rețelele ce conțin bobine sînt din ce în ce mai utilizate datorită rezultatelor superioare ce se obțin și posibilității de reglaj pe care le oferă bobinele. Astfel se poate obține curba optimă, pentru un răspuns în frecvență cît mai bun și distorsiuni neliniare cît mai mici.

Conectarea amplificatorului de înregistrare la capul de înregistrare se face (fig. 6) printr-o rezistență de valoare mare (zeci de kΩ) pentru a simula generatorul de curent, în serie cu un circuit rezonant LC ce atenuează semnalul de premagnetizare ce ar putea pătrunde în amplificatorul de înregistrare. Uneori acest circuit LC poate lipsi. De asemenea, dacă etajul final al amplificatorului de înregistrare are impedanță mare de ieșire (ca în cazul schemei din fig. 5e), rezistența R poate avea valori mai mici (kΩ). Tot pe capul de înregistrare se aplică și curentul de polarizare provenit de la oscilatorul de ștergere, prin grupul R_1C_1. Cu R_1 se reglează valoarea optimă a curentului pentru distorsiuni minime și bandă de frecvență maximă. Între aceste două cerințe trebuie făcut un compromis, acceptîndu-se o bandă de frec-

Amplificatoare de înregistrare

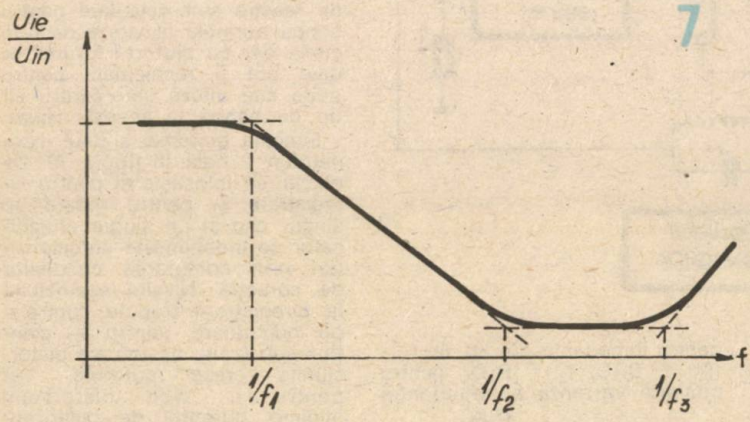


Conectarea amplificatorului la capul de înregistrare

6

Curba de răspuns a amplificatorului de redare

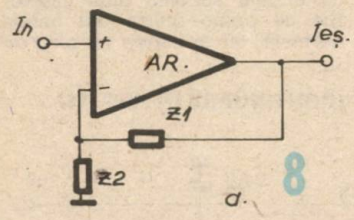
7



curentul de premagnetizare astfel ca o frecvență de 10 kHz să se înregistreze cît mai tare, apoi se mărește curentul cu încă 2-4 dB (de 1,3 — 1,6 ori mai mare).

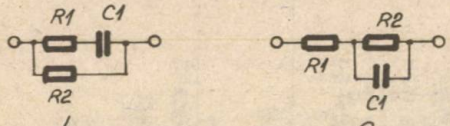
La redare, variațiile de flux provocate de micșorarea benzii prin fața capului de redare induc în acesta o tensiune electromotoare ce e amplificată și apoi redată într-un difuzor. La redare, caracteristica amplitudine/frecvență prezintă un minim în zona frecvențelor joase, atinge un maxim la frecvențe medii, apoi scade rapid la frecvențe înalte. Este evidentă necesitatea introducerii unor corecții și în amplificatorul de redare. Forma curbei la redare (fig. 7) are următoarele caracteristici: panta e de 6 dB/octavă (deci se folosesc rețele simple RC) și are 3 timpi: t_1, t_2, t_3, din care t_1 și t_2 sînt standardizați în funcție de viteza și tipul benzi-

venţă ceva mai redusă, dar avînd avantajul unor distorsiuni sensibil mai mici. Reglarea se poate face destul de ușor: se reglează



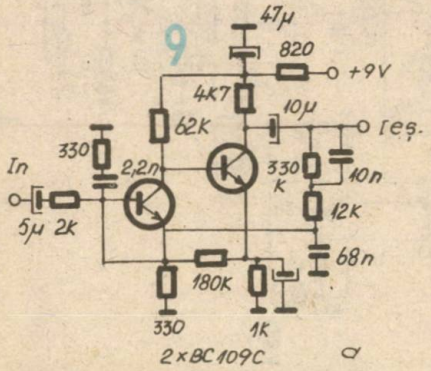
8

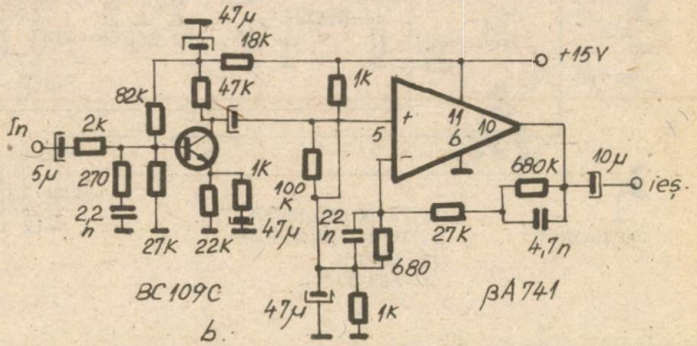
Viteza	Tipul benzii	t_1	t_2
19,05 cm/s		3180 μs	50 μs
9,53 cm/s		3180 μs	90 μs
4,76 cm/s	normal	3180 μs	120 μs
	FeCr	3180 μs	90 μs
	CrO_2 Metal	3180 μs	70 μs



lor folosite; t_3 nu e standardizat, el poate lipsi la viteze mari ale benzii. Acești timpi caracterizează punctele în care curba își schimbă direcția; conform normelor RIAA sînt dați în tabelul de mai sus.

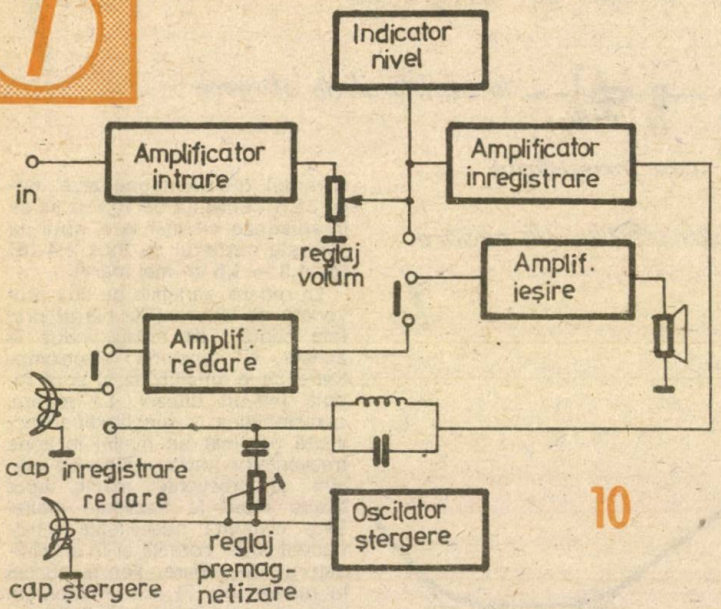
9





7

Schema bloc a unui magnetofon



10

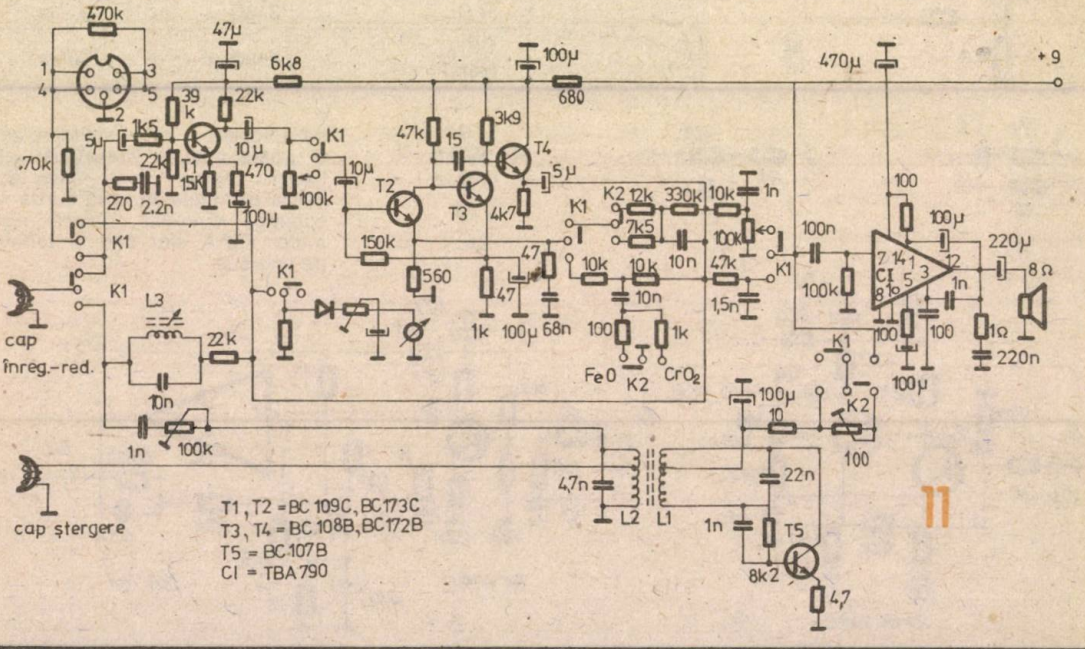
Circuitele de corecție la redare pot fi cuplate între etajele amplificatorului de redare sau în bucla de reacție negativă a acestuia. Vom analiza doar acest ultim caz, fiind cel mai des întîlnit, datorită avantajelor pe care le oferă. Schema generală a amplificatorului de redare este dată în figura 8a. Structura impedanței Z_1 poate fi cea din figura 8b sau cea din figura 8c, iar a impedanței Z_2 e dată în figura 8d. Relațiile de calcul sînt următoarele: pentru impedanța din fig. 8b $t_1 = (R_1 + R_2)C$, $t_2 = R_1C_1$, pentru cealaltă variantă a impedanței Z_1, $t_1 = R_2C_1$; $t_2 = \frac{R_1R_2}{R_1+R_2}C_1$. În ambele cazuri $t_3 = R_3C_2$. Dacă C_2 lipsește, nu va exista t_3. Pentru ca relațiile scrise să fie valabile, e necesar ca $Z_1 \ll Z_2$, altfel t_2 își va modifica valoarea. Amplificarea etajelor va fi $A = \frac{Z_1}{Z_2}$.

O altă cerință a amplificatorului de redare e aceea de a asigura un raport semnal-zgomot cît mai mare, adică de a avea zgomot propriu cît mai redus. În acest scop se utilizează tranzistoare cu zgomot redus (BC 109, BC 173), care sînt puse să lucreze în regim de zgomot minim (curent de colector cuprins între 50 și 200 μA, impedanța generatorului de 2—10 kΩ).

În figurile 9a și 9b sînt două variante de scheme practice. Valorile elementelor din rețeaua de reacție sînt calculate pentru banda normală la viteza de 4,76 cm/s, dar cu ajutorul formulelor date pot fi recalculate pentru orice altă viteză sau pentru alt tip de bandă la aceeași viteză.

Schema generală a unui magnetofon e dată în figura 10. De obicei, se folosește și pentru înregistrare și pentru redare un singur cap și un singur amplificator ce îndeplinește ambele roluri prin comutarea circuitelor de corecție. Nivelul semnalului la înregistrare trebuie controlat pe indicatoare, pentru a-l menține sub limita peste care distorsiunile cresc puternic. Tot pentru a avea distorsiuni minime, curentul de polarizare trebuie să fie cît mai perfect sinusoidal, orice armonici superioare ale acestuia afectează calitatea înregistrării. În figura 11 e dată schema unui casetofon ce poate utiliza atît banda normală, cît și banda cu oxid de

(CONTINUARE ÎN PAG. 95)



11

ce știm DESPRE ÎNREGISTRĂRI

Producția benzilor de magnetofon, a casetelor cunoaște o continuă dezvoltare. Pentru a se asigura compatibilitatea benzilor și casetelor cu diverse tipuri de magnetofoane sau casetofoane s-au emis o serie de instrucțiuni cuprinzînd dimensiunile standard sau alte caracteristici ale benzilor de înregistrare. Pentru a veni în sprijinul amatorilor de înregistrări vom oferi în rîndurile ce urmează o serie de date în ceea ce privește benzile, precum și diverse posibilități de trucaj sonor.

Timpul indicat este mediu. El poate varia în funcție de viteza exactă a magnetofonului. Timpul este valabil pentru audiția unei benzi stereo înregistrate pe două piste. El trebuie dublat pentru audiția unei înregistrări mono pe două piste și pentru înregistrările stereo pe 2 x 2 piste. Timpul se va multiplica cu 4 pentru înregistrarea mono pe 4 piste.

Exemplu: o bandă cu durată dublă montată pe o bobină de 18 cm înregistrată mono cu viteza de 4,75 cm/s permite 960 minute sau 16 ore de audiție (vezi tabelul 2 și apoi tabelul 3).

Durata audiției în casete Compact înregistrate mono sau stereo este: C60 – 2 x 30 minute; C90 – 2 x 45 minute; C120 – 2 x 60 minute.

Suportul poate fi: PVC — clorură de polivinil; CA — acetat de celuloză; PEPT — poliester.

Benzile în poliester sînt cunoscute curent și sub forma de bandă Mylar. Un suport mai recent este Tenzar, ce poate fi clasificat în categoria poliesterului.

Fig. 1a. **Poziția pistelor și a sensului de derulare indicate în normele de standardizare (utilizate de anumiți producători).**

Fig. 1b. **Poziția pistelor și a sensului de derulare indicată în reviste, în literatura de specialitate și în majoritatea cataloagelor.**

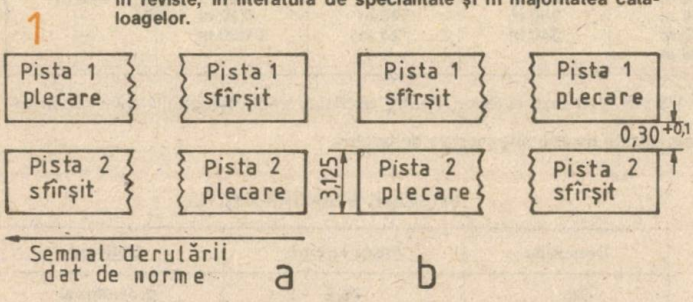
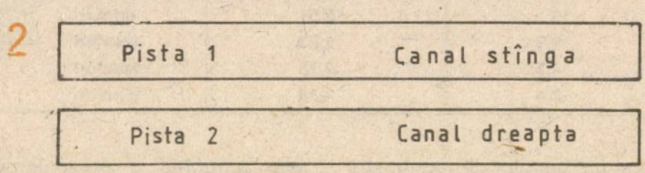


Fig. 2. **Pentru înregistrări stereofonice pista 1 este canalul stînga și pista 2 este canalul dreapta (DIN 45.511).**



MONTAREA AMORSELOR

Inceput de bandă: amorsă verde 1 m lungime. Sfîrșit de bandă: amorsă roșie 1 m lungime. Există benzi amorsă de culoare bleu sau galbenă.

Pentru a interpreta desenele normale de standardizare a poziției pistelor și sensurilor de defilare a benzii, trebuie să ne imaginăm că ochiul observatorului este în locul capului magnetic (adică banda este văzută dinspre partea oxid). Însă în literatura tehnică și deseori în cataloage se consideră că ochiul observatorului se află unde este în

DIMENSIUNI, TOLERANȚE, DENUMIRI

LĂȚIME	TOLERANȚA LA LĂȚIME	GROSIME TOTALĂ	DENUMIRE
6,25 mm	±0,06	48 microni	normală
6,25 mm	±0,06	35 microni	durată lungă
6,25 mm	±0,06	20 microni	durată dublă
6,25 mm	±0,06	18 microni	durată triplă
6,25 mm	±0,06	13 microni	durată cvadruplă
			Casete
3,81	±0,05	25 microni	C60
3,81	±0,05	18 microni	C90
3,81	±0,05	13 microni	C120

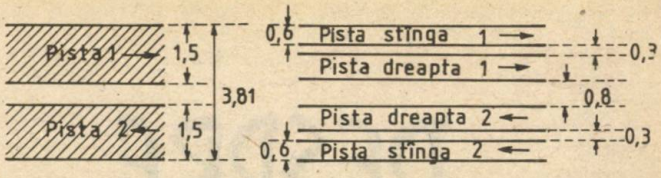


Fig. 4. **Poziția pistelor în casete Compact. Se remarcă faptul că în înregistrările stereo pistele nu sînt încrucișate. Sensul derulării indicat corespunde normelor, fiindcă în casete banda este rulată cu oxidul în exterior.**

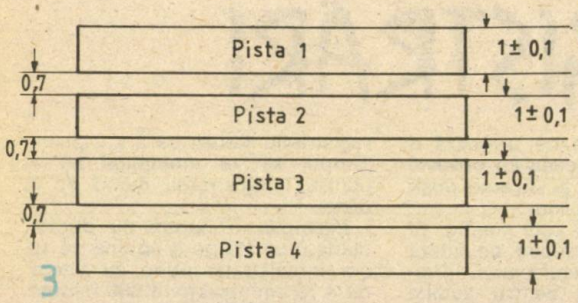


Fig. 3. **Pentru înregistrări stereo pista 1 lucrează cu pista 3 și pista 2 cu pista 4.**

Pista 1 canal stînga
Pista 3 canal dreapta } **prima înregistrare**

Pista 2 canal dreapta
Pista 4 canal stînga } **a doua înregistrare**

CONȚINUTUL BOBINELOR ÎN FUNCȚIE DE GROSIMEA BENZII

Diametrul bobinei	Diametrul central[1]	Normală	Durată lungă	Dublă durată	Triplă durată	Cvadruplă durată
8 cm	45 mm	—[2]	65 m	90 m	135 m	180 m
10 cm	45 mm	90 m	135 m	180 m	270 m	360 m
11 cm	45 mm	—	—	270 m	360 m	540 m
13 cm	45 mm	180 m	270 m	360 m	540 m	720 m
15 cm	50 mm	270 m	360 m	540 m	720 m	—
18 cm	60 mm	360 m	540 m	720 m	1 080 m	—
25 cm	70 mm	760 m	—	—	—	—

[1] **În principiu, diametrul central este 1/3 din diametrul exterior. Pentru bobinele mici diametrul minim recomandat de norme este de 45 mm.**

[2] **Anumite tipuri de benzi nu sînt comercializate pe anumite modele de bobine.**

realitate. În casetofoane, unde în principiu ochiul observatorului se găsește în partea suportului (fiindcă banda în casete este realizată cu partea oxid în interior), sensul defilării indicat este invers celui dat în norme. O asemenea metodă antrenează o serie de neclarități. Dacă insistăm asupra acestei probleme o facem fiindcă anumite cataloage sînt redactate conform normelor oficiilor de standardizare.

Există benzi etalon ce vor permite constructorilor și depanatorilor verificarea aparatelor atît la înregistrare, cît și la redare. Aceste benzi sînt indispensabile pentru azimutajul capetelor magnetice și se va vedea din tabel cît de sever este acest test pentru casetofoane.

VITEZELE STANDARDIZATE

Denumire	Viteză în cm/s	Utilizare
76	76,2	profesional
38	38,1	profesional
19	19,5	profesional
19	19,5	amatori
9,5	9,53	amatori
4,75	4,76	amatori
2,4	2,38	amatori

TRUCAJE — DEFINIȚII

Duo-play. Acest procedeu permite înregistrarea unui program pe o pistă în timp ce este ascultată o înregistrare efectuată pe altă pistă. Cele două înregistrări sînt sincronizate fiindcă întrefierurile capului de înregistrare și ale capului de redare sînt pe aceeași axă. La redare ambele capete citesc, deci cele două înregistrări sînt sincronizate.

Multiplay. Se realizează prin extensia procedeului anterior; înregistrarea făcută pe prima pistă este mixată cu a doua înregistrare. În acest caz, se ascultă prima înregistrare, dar se transferă pe noua pistă pe care înregistrează al doilea program. Cele două programe sînt astfel înregistrate pe o singură pistă. Acest procedeu de transfer al unui program de pe o pistă pe alta adăugînd o nouă înregistrare poate fi repetat de trei, patru, cinci ori chiar, permițînd astfel unui interpret să înregistreze toate partiturile unei mici formații.

Contrar procedeului precedent, întrefierurile nu au nevoie de aliniere, fiindcă pista originală nu servește la nimic. În principiu, toate aparatele stereo cu cap separat de înregistrare și redare permit înregistrări multiplay.

Reverberația. Toate aparatele mono și stereo cu capetele de înregistrare și redare montate

separat permit realizarea efectului de reverberație dacă cele două capete sînt suficient de apropiate și dacă se utilizează viteza 19 cm/s (la 38 cm/s rezultatele sînt mai bune). Pentru a realiza aceasta se reintroduce semnalul de la redarea înregistrării în curs de efectuare cu ajutorul unei cutii de mixaj.

Dacă magnetofonul nu are cutie de mixaj, se pot obține rezultate excelente citind banda la nivel foarte slab cu un difuzor plasat la cîțiva metri de microfon.

Ecoul. Efectul se realizează foarte ușor cu aparate mono și stereo cu capete de înregistrare și redare separate. Se pot folosi metodele preconizate pentru efectele de reverberație. Pentru obținerea ecoului sînt utilizate vitezele 9,5 cm/s și 4,75 cm/s.

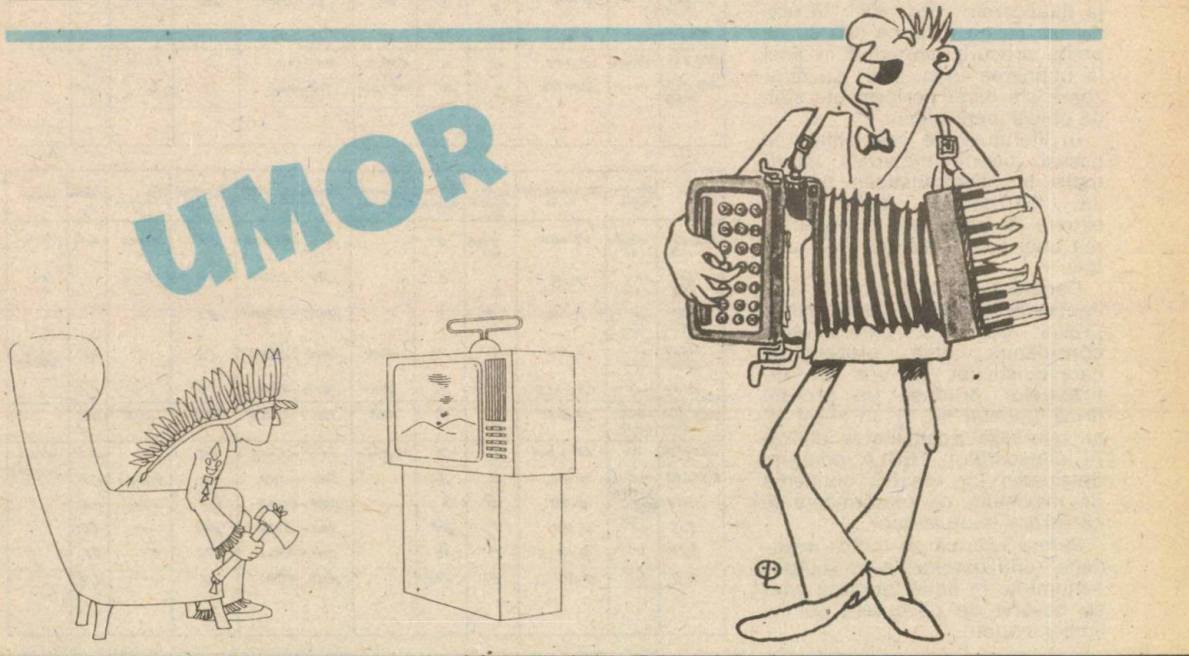
Cu potențiometrul circuitului de ascultare se poate mări sau micșora numărul de repetiții.

DURATA TIMPULUI DE AUDIȚIE A UNEI PISTE ÎN FUNCȚIE DE LUNGIMEA BENZII ȘI VITEZĂ (în minute)

Viteza / Lungimea	19 cm/s	9,5 cm/s	4,75 cm/s	2,4 cm/s
45 m	3,75	7,5	15	30
65 m	5,5	11	22	45
90 m	7,5	15	30	60
135 m	11	22	45	90
180 m	15	30	60	120
270 m	22	45	90	180
360 m	30	60	120	240
540 m	45	90	180	360
720 m	60	120	240	480
1 080 m	90	180	360	720

PROPRIETĂȚILE BENZILOR ETALON

	Viteza în cm/s	Nivel de referință	Frecvența semnalului de referință Hz	Plaja de răspuns în frecvență de la... la... Hz	Plaja de răspuns în frecvență dB
Bandă etalon DIN 19,1/4	19,05	32	1 000	31,5/18 000	—20
Bandă etalon DIN 9,1/4	9,53	25	333	31,5/16 000	—20
Bandă etalon DIN 4,75 1/4	4,75	25	333	31,5/10 000	—30



7

INCINTE ACUSTICE

Ing. AURELIAN MATEESCU

În materialul de față revenim la considerațiile prezentate în articolul „Incinte acustice", apărut în almanahul Tehnium '85 (pag. 58—63). Se cunoaște faptul că incintele acustice reprezintă veriga slabă a lanțului de reproducere a sunetului, fiind elementele cu randamentul cel mai scăzut, cu fiabilitatea cea mai mică și cu fidelitatea cea mai redusă a lanțului electroacustic. Pentru constructorul amator construirea unor incinte acustice de calitate reprezintă deseori o problemă greu solubilă datorită necunoașterii parametrilor tehnici ai difuzoarelor de care dispune, folosirii de materiale nepotrivite, adoptării de soluții improprii. De cele mai multe ori, constructorul amator ajunge la soluția de a copia o incintă acustică de producție industrială în ceea ce privește „cutia" propriu-zisă și uneori elementele rețelei de separare a frecvențelor, dar, neavînd acces la difuzoarele „originale", le realizează cu ce dispune sau cu ce poate procura, ajungînd în final la obținerea unei „lăzi" cu difuzoare ale cărei performanțe sînt, de obicei, deplorabile.

În literatura de specialitate se găsesc puține materiale referitoare la caracteristicile tehnice ale difuzoarelor, caracteristici extrem de necesare la proiectarea unei incinte acustice de calitate.

Deoarece în țara noastră sînt importate produse ale firmei „Tesla" din R.S. Cehoslovacă, considerăm utilă prezentarea caracteristicilor tehnice ale difuzoarelor produse de această firmă (tabelul nr. 1). În acest fel se ușurează posibilitatea utilizării difuzoarelor TESLA conform destinației lor, pentru obținerea de maximum de performanțe în constructiile de amator.

Pentru rubrica ce indică destinația difuzoarelor s-au utilizat denumirile în limba engleză care au devenit de uz curent pentru limbajul audio:

woofer — difuzor construit special pentru redarea frecvențelor joase;

subwoofer — difuzor pentru redarea frecvențelor foarte joase;

middrange — difuzor pentru redarea frecvențelor medii;

tweeter — difuzor pentru redarea frecvențelor înalte. Din construcție acestea sînt întîlnite în literatura de specialitate sub denumirile următoare:

a) cone tweeter — difuzor pentru frecvențe înalte cu membrana con;

b) dome tweeter — idem, cu membrana în formă de calotă sferică;

c) bullet radiator tweeter — idem, cu membrana radiantă în formă asemănătoare unui vîrf de glonț.

În tabelul nr. 2 sînt cuprinse caracteristicile difuzoarelor pentru frecvențe joase (woofer) produse în R.P. Bulgaria.

Din experiența căpătată în urma construcției mai multor tipuri de incinte acustice, în mai multe variante și cu difuzoare de proveniență foarte diversă, am desprins următoarele concluzii extrem de folositoare atunci cînd se dorește obținerea unui maxim de calitate acustică:

— nu se obțin rezultate bune

Tabelul 1. — Difuzoare TESLA (R.S.C.)

Tipul	Destinația	Dimensiuni mm	Impedanța (Ω)	Puterea nominală W	Frecvența rezonanță Hz	Banda de frecvențe reprodusă Hz	Presiunea acustică dB	Tip magnet	Masa kg	Obs.
ARN5604 5608	woofer	⌀ 165	4 8	15	35	40 — 4000	80	ferită	1	
6604 6608	-"- -"-	⌀ 203	4 8	20	29	40 — 4000	89	-"-	1,1	
8604 8608	-"- -"-	⌀ 312	4 8	30	20	35 — 1000	89	-"-	4,5	
932	-"-	⌀ 390	15	25	38	38 — 1000	92	Alnico	10,5	
ARO 935	subwoofer	⌀ 340	4	10	30	30 — 4000	96	Alnico	5,51	
932	-"-	⌀ 390	8	15	24	24 — 4000	98	-"-	10,2	
962	-"-	⌀ 390	30	15	24	24 — 4000	98	-"-	10,2	
ARM9304 9308	-"-	⌀ 388	4 8	50	70	50 — 5000	100	-"-	8	
9404 9408	-"-	⌀ 390	4 8	50	30	42 — 1000	100	-"-		
9315	-"-	⌀ 390	15	50	30	42 — 1000	100	-"-		
ARO 664	mid-range	⌀ 203	15	5	60 ÷ 95	100 — 6000	90	ferită	0,34	
ARZ 4604 4608	-"-	123 × 123	4 8	20	235 ÷ 250	500 — 4000	89	-"-	0,85	

Tipul	Destinația	Dimensiuni mm	Impedanța Ω	Puterea nominală W	Frecvența rezonanță Hz	Banda de frecvențe reprodusă Hz	Presiunea acustică dB	Tip magnet	Masa kg	Obs.
ARV 884 888	tweeter	75 × 50	4 (5,5) 8	2		1000 ÷ 16.000	87	Alnico	0,18	eliptic
164 168	-"- -"-	⌀ 90	4 8	5		1500 ÷ 20.000	92	ferită		con -"-
160 167	-"- -"-	75 × 50	15 8	5		2500 ÷ 20.000	92	-"-		-"- -"-
3604 3608	-"- -"-	⌀ 120	4 8	10	1200	2000 ÷ 20.000	89	-"-	0,75	calotă (dome)
1304	-"-	57,5 × 57,5	4	10	1500	1800 ÷ 20.000	87	-"-	0,17	
ARX 364 368	banda largă	⌀ 103	4 8	5	100	100 ÷ 15.000	86	ferită	0,17	
ARV 4704	-"-	130 × 130	4	15	50 ÷ 60	45 ÷ 15.000	88	-"-	0,5	plastic aluminiu
ART 382	middrange banda medie largă	⌀ 82	15	3		500 — 4000	90	Alnico	0,7	
383	-"-	⌀ 108	15	3		500 — 4000	92	-"-	0,5	
981	-"-	⌀ 90	8	45		300 — 3500	113	-"-	1,6	
983	-"-	⌀ 90	8	45		300 — 3500	113	-"-	1,6	
150	-"-	⌀ 100	8	150		250 — 8500	111	-"-	2,2	

Tabelul nr 2. DIFUZOARE de JOASĂ FRECVENȚĂ (WOOFERE) producție R.P. BULGARIA

Tip	P_n/P_n^* W	Impedanța Z (Ω)	Frecv. rezonanță Hz	Frecv. superioară Hz	Melimanitate dB	Dimensiuni gabarit mm	Masa Kg	Observații
BK 138A	4/15	4,8	80±10	12.500	12	132 × 132 × 84	0,950	P_n^* – puterea nominală pentru frecvențe pînă în 200 Hz
BK 138Б	4/15	4,8	80±10	12.500	12	132 × 132 × 84	0,950	
BKH0822	20/30	4	60±10	4.000	12	125 × 125 × 60	0,980	
BKH0832	20/30	8	60±10	4.000	12	125 × 125 × 60	0,980	
BKH0922	20/30	4	50	5.000	12	160 × 160 × 75	1	
BKH0932	20/30	8	50	5.000	12	160 × 160 × 75	1	
BBK20164	10/50	4	40	4.000	12	208 × 208 × 85	1,2	
BBK20188	10/50	8	40	4000	12	208 × 208 × 85	1,2	
BKH1231	40	8	25±5	4000	12	⌀ 312 × 158	3,6	
BKH1233	80	8	25±5	4.000	12	⌀ 315 × 158	4	

prin copierea produselor de serie (industriale) atunci cînd nu dispunem de aceleași difuzoare.

În plus orice produs industrial este un compromis între preț și calitate; or, constructorul amator poate obține rezultate mai bune dacă depune un plus de muncă și poate obține un rezultat foarte bun din punct de vedere acustic;

— de calitatea execuției tîmplăriei și montarea etanșă a difuzoarelor depind enorm calitatea incintei, fiabilitatea ei și a difuzoarelor;

— se va păstra totdeauna destinația difuzoarelor pentru obținerea de bune rezultate. Un difuzor woofer pentru incinte închise va reda distorsionat și se va distruge rapid în incinte deschise (tip bassreflex);

— se va prefera calculul incintei și al rețelei de separare a difuzoarelor soluției de a prelua niște valori ale unei soluții existente, dar care lucrează cu alte difuzoare (pot diferi diametrele difuzoarelor, impedanța, frecvența de rezonanță etc.).

BIBLIOGRAFIE:

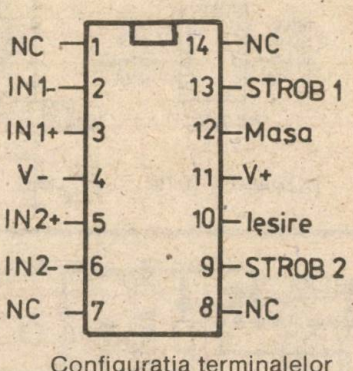
Colecția R.T.E., 1985, R.P. Bulgaria

Colecția Amaterske Radio, 1984, R.S.C.

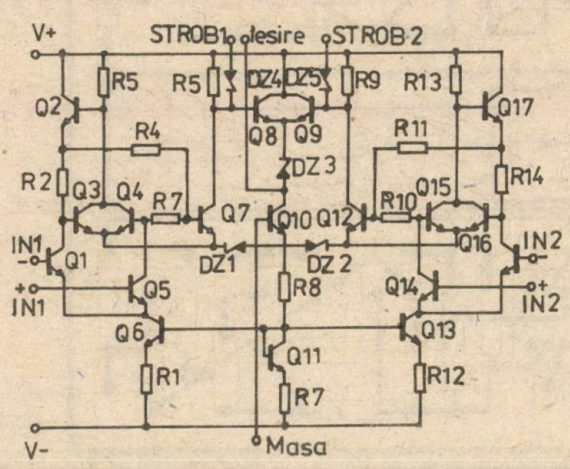
CLB 2711 EC

Circuitul CLB 2711 EC conține două comparatoare de tensiune cu intrări diferențiale, separate, ieșire comună și intrări de strobare independente pentru fiecare canal.

Cu o rețea rezistivă externă pot fi utilizate ca amplificatoare de lectură pentru memorii cu ferită. Se mai pot folosi ca detector dublu-canal cu performanțe superioare celor obținute prin conectarea a două comparatoare.



Configurația terminalelor



(URMARE DIN PAG. 90)

crom. Amplificatorul de înregistrare-redare e construit cu tranzistoarele T_1—T_4. Comutatorul K_1 (pe poziția redare) comută corecțiile pentru înregistrare sau redare, iar K_2 e folosit la schimbarea tipului de bandă. Bobina oscilatorului (construit cu T_5) se bobinează pe o oală de ferită. L_1 are 2 x 32 spire, iar L_2 140 spire cu priza la spira 42 din sîrmă de cupru emailată de 0,1 mm. Bobina L_3 din circuitul rezonant paralel pentru blocarea semnalului de polarizare se construiește pe o carcasă de FI utilizată în receptoare și are 120 spire din același tip de sîrmă. Frecvența oscilatorului e de circa 80 kHz. Prin scurtcircuitarea semireglabilului de 100 Ω e mărită tensiunea de alimentare a oscilatorului și, implicit, e mărit curentul de polarizare și curentul de ștergere pentru banda cu dioxid de crom. Amplificatorul de audiofrecvență e construit cu circuitul integrat TBA790.

BIBLIOGRAFIE:

B. Barbat, I. Presură, T. Tănăsescu — Amplificatoare de audiofrecvență

Revista „Radio" (U.R.S.S.)

7

PREAMPLIFICATOR PENTRU PICUP NC430

Student FLORENTIN LUCACI

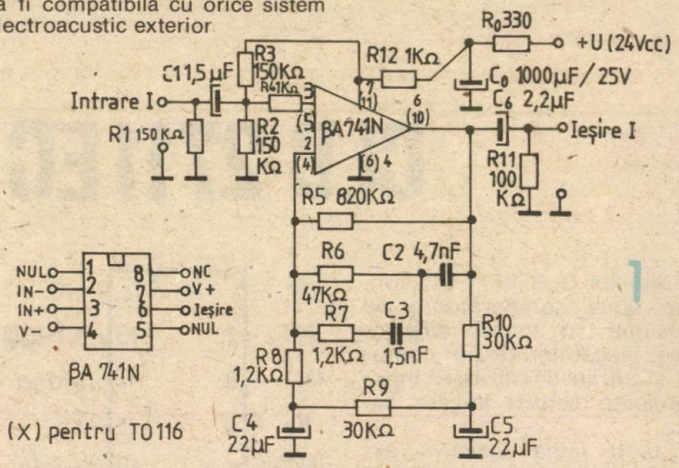
În figura 1 este prezentată schema de principiu (CANALUL I) a preamplificatorului. Semnalul este injectat pe intrarea neinversoare (+) a circuitului operațional βA 741 N prin condensatorul C_1. Amplificarea globală este neliniară, obținîndu-se o funcție de transfer ce urmărește foarte bine caracteristica A(f) a normelor RIAA. Alimentarea este obișnuită, făcîndu-se de la o sursă de tensiune bine filtrată. Condensatoarele de compensare externă C_2, C_3 sînt cu stiroflex, iar cele de cuplaj C_1, C_6 cu tantal. Se vor folosi rezistențe peliculare de 0,25 ÷ 0,5 W, în special în cazul rezistoarelor R_1, R_2, R_3, R_4, R_5, R_6, R_7. Conexiunile între picupul NC 430 și montaj și mai departe la amplificator se fac cu cablu ecranat.

În figura 2 este prezentat cablajul imprimat (CANALUL I · CANALUL II) la scara 1:1. Acestuia nu i se va aduce nici o modificare constructivă, fiind proiectat pentru evitarea oricăror factori paraziți. Ca material este de preferat sticlotextolitul. Montajul se poate lăsa și neecranat neinfluențîndu-se ușor de la factorii exteriori.

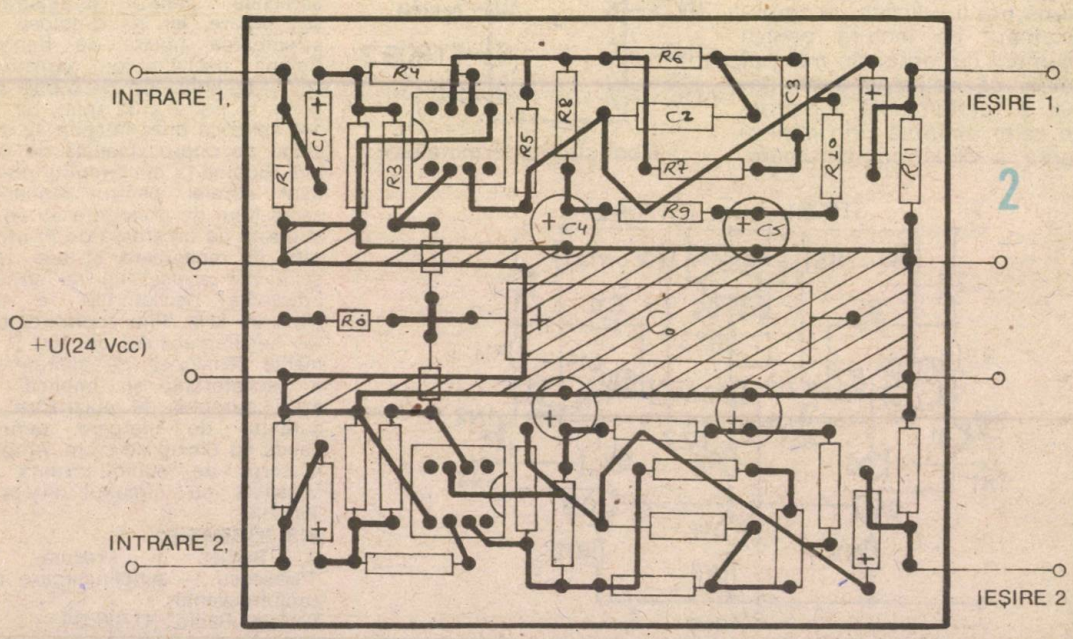
Constructorii mai experimentați pot încerca amplasarea lui în interiorul picupului NC 430, împreună cu sursa de alimentare. În cazul acesta borna de ieșire va fi compatibilă cu orice sistem electroacustic exterior

CARACTERISTICI:

— intrare: doza VM2102 (sau alte tipuri, fără nici o modificare);
— $20 \leq f \leq 20\,000$ Hz;
— dinamică;
— diafonie (a dozei).



1



2

CONDUCEREA PREVENTIVĂ

Dr. ing. TRAIAN CANȚĂ

Conștient sau inconștient, din momentul urcării la volanul unui autovehicul — indiferent de tipul sau categoria lui —, conducătorul auto este supus unor riscuri mai mari sau mai mici. Excluzînd de la început ideea „fatalității" în petrecerea unui eveniment (accident) rutier, după datele statistice, se poate afirma că valoarea **conducătorului autovehiculului este factorul cel mai important care decide de cele mai multe ori. După cum se va vedea în continuare, în acest factor complex intră o serie întreagă de calități native sau care se dobîndesc pe parcurs, prin educație.**

Există o opinie generală prin care se afirmă că abia după parcurgerea a 100 000 km un conducător auto poate conduce bine și „fără probleme", prin aceasta considerîndu-se că în perioada de timp respectivă conducătorul auto a avut prilejul să treacă prin situații mai mult sau mai puțin grele, situații care, de fapt, reprezintă un antrenament continuu spre perfecționare.

Se poate accepta acest lucru, cu oarecare reținere, deoarece, în cazul fiecărui individ, mai intervin și calitățile personale, o cultură tehnică, o anumită educație rutieră. După cum am văzut anterior, această „experiență la volan" este, de fapt, reflexul unui cumul de evenimente trăite, petrecute în timpul îndelungatei perioade de conducere auto.

S-a constatat că rezolvarea favorabilă a acestor „evenimente" este legată direct de calitățile personale ale individului, de stilul său de conducere, de nivelul agresivității comportamentale, precum și de riscurile ivite și chiar, într-o mai mică măsură, de șansă.

Experiența la volan poate fi dobîndită și fără a rula sute de mii de kilometri, efectuînd pentru aceasta antrenamente în condiții diferite de rulare.

În Statele Unite ale Americii, de exemplu, există școli de conducere pentru începători, unde, pentru formarea rapidă a reflexelor, pentru perfecționarea tehnicii de conducere, în timpul antrenamentelor, viitorilor conducători auto li se simulează diferite situații nepreväzute, similare cu cele întîlnite pe căile rutiere, cum ar fi de exemplu: apariția neașteptată a unui om (la nivelul solului, printr-o trapă este ridicată la cîțiva metri în fața mașinii o figură din carton — care, evident, trebuie evitată), crearea instantanee a unor porțiuni de drum umed sau cu noroi, la viteze ridicate și în curbe ș.a.m.d. Aceste antrenamente se fac după un ciclu, cu etape tot mai grele, stabilite științific, fiind conduse direct de la un pupitru de comandă, situat undeva la înălțime, pentru a domina și conduce întregul poligon.

În timpul participării la circulația rutieră, fiecare dintre noi acceptă anumite riscuri și renunță la altele. Chiar și cei care nu conduc autovehicule, dar care sînt permanent în contact cu fluxul rutier, cunosc factorii de risc în conducerea automobilului, care pot fi grupați în cîteva mari categorii: starea tehnică a autovehiculului (sistemul de frînare, direcția, pneurile etc.), starea vremii (polei, ceață, viscol ș.a.), caracteristicile fiziologice și psihologice ale celui de la volan (oboseala, starea bahică, lipsa de experiență, agresivitatea la volan), precum și nivelul cunoștințelor personale, specifice circulației rutiere (însușirea pînă la obișnuință a prevederilor regulamentului de circulație, cunoașterea posibilităților autovehiculului pe care-l conduce, posedarea unor deprinderi comportamentale în conducere față de starea și configurația traseului etc.).

Privind **agresivitatea comportamentală** a indivizilor, care — foarte curios — capătă **valențe superioare** atunci cînd persoana în cauză se află **la volan,** valențe amplificate poate de „caii putere" pe care îi strunește, aceasta ia forme dăunătoare, periclitează securitatea circulației rutiere și conduce, în general, la pagube materiale și la pierderi de vieți omenești.

Ca urmare a unor studii efectuate de către diferiți cercetători și psihologi, acest mod de comportare, denumit de noi **„agresivitate la volan",** este întîlnit numai în anumite cazuri, prezentate în continuare: alcoolicii, tinerii, ignoranții, vitezomanii și alergătorii de curse.

Cei mai periculoși sînt considerați **alcoolicii,** care, sub influența băuturilor consumate, devin mai întîi imprudenți, pentru ca, pe măsură ce se „adaptează" condițiilor de circulație, să devină temerari și, în același timp, extrem de periculoși.

Acest stil „obraznic" de a rula, din nefericire, devine foarte agresiv față de ceilalți parteneri la fluxul rutier. Practica a demonstrat că, în majoritatea acestor cazuri, finalul este ori tragic, ori cu pagube materiale însemnate.

Normal că este indicat a se face cele mai mari eforturi de către cei în drept, precum și de opinia publică, pentru a se evita cooptarea în circulația rutieră a unor astfel de participanți la conducerea automobilelor, purtători ai unei atît de dăunătoare patimi. Este de neînțeles cu cîtă delăsare și indiferență se manifestă cei care participă alături de astfel de conducători auto la petreceri, nunți, revelioane ș.a.m.d., fără a lua nici o atitudine.

Influența alcoolului asupra comportamentului celui în cauză este graduală, perioada respectivă putîndu-se deosebi prin două faze: prima, după consumarea a 1—2 pahare, de exemplu, de vin sau bere, este foarte înșelătoare, în urma căreia conducătorul auto se simte destul de bine, capătă mai mult curaj față de normal, este volubil și deseori devine „fericit", pozînd în mare conducător de automobil, spre admirația celor din anturajul său; a doua fază — care este o continuare a primei — devine deseori tragică, deoarece, în majoritatea cazurilor, bazîndu-se pe experiență (evident, stimulată mult de cantitatea de alcool ingerată), individul în cauză, fără a mai judeca consecințele faptelor sale, devine „puternic" și agresiv, intrînd, fără noimă, pe căile rutiere circulate atît de pietoni, cît și de autovehicule.

Din această cauză, asprimea legilor în vigoare este salutară, tendința firească fiind de a exclude din marea familie a automobiliștilor această categorie de indivizi.

A doua grupă de conducători auto agresivi și deci periculoși este aceea a unor **tineri** care de-abia au primit permisul de conducere și care caută să facă pe „vitejii" în fața celorlalți prieteni din grup, bravînd în postura de „stăpîni ai volanului" și, drept urmare, încalcă regulile de circulație, aleargă nebunește și cu viteze mari, deseori sfidînd riscurile care apar datorită stării și configurației drumului, rezultatul final fiind, deseori, catastrofal, mai ales dacă această „bravadă" de începător este combinată și cu mai puțin sau mai mult alcool.

Ignoranții — o altă categorie de agresivi — sînt conducătorii auto începători, indiferent de vîrstă, care în special cînd circulă „afară", adică pe căile rutiere interurbane, la deplasarea în concediu, la mare sau la munte, sau cu ocazia diferitelor sărbători, rulează cu viteze mult mai mari atît față de prevederile regulamentare, cît și față de configurația și starea drumurilor, efectuează depășiri riscante etc. Evident, aceste manevre depășesc posibilitățile unor astfel de ignoranți. În această categorie mai intră și cei care-și cumpără pentru prima oară autoturism; mai ales dacă acest autoturism este de ultim tip, cum ar fi, de exemplu, automobilele Oltcit. Tot în această categorie intră și acei posesori de autoturisme noi (printre care și Oltcit) care, chiar dacă au o experiență mai mare la volan, nu cunosc bine posibilitățile autoturismului.

Nimeni nu poate să-i accepte pe cei care, rulînd în astfel de condiții, depășesc inoportun celelalte vehicule, gîndind și sperînd că... va frîna celălalt conducător auto, care vine din sens opus, deoarece n-o să-și riște integritatea mașinii și chiar viața ?? Și... dacă o dată și o dată se întîlnesc doi indivizi cu aceeași mentalitate?

Se întîmplă ceea ce vedem aproape zilnic: șosele cu pomi rupți, autovehicule transformate în mormane de fiare și — ceea ce este mai tragic — foști oameni transformați în „masă de carne" sau — în cazuri salutare — oameni accidentați mai mult sau mai puțin grav, care cu greu sau niciodată nu se vor mai putea reface fizic și moral.

O altă categorie de agresivi o reprezintă **maniacii vitezei,** de fapt acea mică parte dintre conducătorii auto care nu admit să fie depășiți. Aceștia rulează pe magistrală sau acolo unde drumul o permite cu viteze nebunești, care, în cazuri extreme, conduc sigur la accidente foarte grave.

Tot pe drumuri cu două benzi de circulație pe fiecare sens, sau pe magistrală, o mică parte dintre conducătorii auto, de asemenea orgolioși, rulează corect, cu viteza de 80 km/oră, dar pe banda de lîngă axa căii rutiere și nu admit să fie depășiți, chiar dacă sînt claxonați sau atenționați cu farurile, indiferent dacă vehiculele care vin din urmă și vor să-i depășească sînt autoturisme oficiale, străine sau proprietate de stat.

Aceasta este, pînă la urmă, tot o problemă de educație, de civilizație rutieră, care nu se mai întîlnește de mult la cei mai avansați (cu mai mulți kilometri la bord), care caută să evite complicațiile, să nu și le creeze singuri.

Pentru că, exact în clipa în care „orgoliosul" se hotărăște să „cedeze" și virează la dreapta, urmăritorul se poate decide, exasperat, să depa-